Muhammad Safi

GNSS Timing Spoofing Detection

Methods and Analysis using Jammertest data

School of Technology and Innovations
Master's thesis
Sustainable and Autonomous Systems

Vaasa 2025

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Muhammad Safi
Title of the thesis: GNSS Timing Spoofing Detection: Methods and Analysis using Jammertest data
Degree: Master of Science in Computing Sciences
Discipline: Sustainable and Autonomous Systems
Supervisor: Heidi Kuusniemi
Evaluator: Mahmoud Elsanhoury
Year: 2025 **Pages:** 105

ABSTRACT:

This thesis investigates GNSS timing spoofing detection strategies using data collected from Jammertest 2024, addressing critical vulnerabilities in infrastructure systems dependent on precise timing. Following a comprehensive literature review of GNSS fundamentals and existing detection methodologies, the research analyses how various parameters behave during spoofing events, including pseudoranges, carrier phase, Doppler measurements, positioning coordinates, signal quality indicators, and HDOP values. Through detailed examination of u-blox F9P receiver data during controlled spoofing events, distinct signature patterns were identified in multiple parameters. While carrier-to-noise ratio monitoring proved ineffective for detection, pseudorange RMS error analysis and NMEA validity flags successfully identified timing anomalies despite generating false positives. The implemented Isolation Forest algorithm demonstrated excellent performance with 100% recall and 99.96% specificity, correctly identifying all spoofing instances while producing only two false positives from 4,695 normal samples. A combined approach using validity flags as initial triggers followed by machine learning verification in the interference data emerged as an optimal strategy, providing a robust framework for protecting critical infrastructure systems against timing attacks.

KEYWORDS: GPS Security, Time Spoofing Detection, Isolation Forest Algorithm, Machine Learning Detection, Resilient Navigation, Critical Infrastructure Protection, GNSS Parameter Analysis, Autonomous Systems

Contents

Figures

Tables

Abbreviations

AGC	Automatic Gain Control
AI	Artificial Intelligence
AMI	Advanced Metering Infrastructure
ANN	Artificial Neural Network
C/N0	Carrier-to-Noise Density Ratio
C1C	Code pseudorange
CART	Classification And Regression Trees
CDMA	Code Division Multiple Access
CEST	Central European Summer Time
D1C	Doppler measurement
dBm	Decibel-milliwatts (power measurement)
DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning

DOA	Direction of Arrival
FDMA	Frequency Division Multiple Access
GGA	Global Positioning System Fix Data (NMEA message type)
GLONASS	Global Navigation Satellite System (Russian)
GNB	Gaussian Naive Bayes
GNGGA	GNSS Fix Data (NMEA message type)
GNRMC	GNSS Recommended Minimum Data (NMEA message type)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HDOP	Horizontal Dilution of Precision
IATA	International Air Transport Association
IEEE	Institute of Electrical and Electronics Engineers
K-means	K-means Clustering
L-SVM	Linear Support Vector Machine
L1C	Carrier Phase
LLS	Lightning Location System
LOS	Line of Sight
LR	Logistic Regression
MANA	NMEA-based Anomaly detection
MiFID	Markets in Financial Instruments Directive
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MW	Megawatt
NavIC	Navigation with Indian Constellation
NCO	Numerically Controlled Oscillator
NMEA	National Marine Electronics Association

OSNMA	Open Service Navigation Message Authentication
PCA	Principal Component Analysis
PNT	Positioning, Navigation, and Timing
PPS	Pulse Per Second
PSU	Phasor Measurement Unit
RF	Random Forest
RL	Reinforcement Learning
RMC	Recommended Minimum Data (NMEA message type)
RMS	Root Mean Square
RXM-MEASX	Receiver Manager Measurement Data (UBX message)
RXM-RAWX	Receiver Manager Raw Data (UBX message)
S1C	Signal strength
SAC	Soft Actor-Critic
SCD-MF	Separate Clock Drift Matched Filter
SCPC	Spoofing Correlation Peak Cancellation
SVM	Support Vector Machine
TD3	Twin Delayed Deep Deterministic Policy Gradient
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TEXBAT	Texas Spoofing Battery
TOA	Time of Arrival
TSA	Time Synchronization Attack
TW	Terawatt
UBX	u-blox Proprietary Protocol
UTC	Coordinated Universal Time
VTG	Course Over Ground and Ground Speed (NMEA message type)
WAMS	Wide Area Monitoring System

1 Introduction

This thesis examines GNSS timing spoofing detection methods and analyses data collected during Jammertest 2024, focusing on both traditional and AI-based detection approaches for critical infrastructure protection.

1.1 Background and Motivation

Global Navigation Satellite Systems (GNSS) provide precise timing services that are essential for critical infrastructure including power grids, telecommunications, financial transactions, and transportation systems. In Finland, where electricity generation is projected to 15 TWh by 2035 (Fingrid, 2024), the energy sector's reliance on precise timing synchronization makes it particularly vulnerable to GNSS disruptions. Apart from grids, even the aviation industry which mostly relies on GNSS positioning is also getting affected by timing spoofing. Recent incidents highlight these vulnerabilities, including a commercial airliner that lost access to onboard digital communication systems due to timing spoofing (Pearson, 2024), while other documented cases show that fuel computation systems and other timing-dependent technologies are similarly susceptible (SKYbrary Aviation Safety, n.d.). As GNSS timing applications continue to expand across critical infrastructure, developing robust spoofing detection and mitigation strategies has become increasingly urgent to protect essential services from sophisticated timing attacks.

1.2 Problem Statement

Despite increasing awareness of GNSS vulnerabilities, effective real-time detection and mitigation of timing spoofing attacks remain challenging. Current research primarily focuses on theoretical models or laboratory settings, with limited field testing of detection methodologies under realistic conditions. This research addresses this practical

implementation gap by analysing empirical and real spoofing data from Jammertest 2024 to develop and validate combined traditional and machine learning approaches for timing spoofing detection, with particular attention to applications in critical infrastructure environments such as Finland's rapidly expanding energy sector.

1.3 Research Objectives

This research aims to achieve the following objectives:

1. To critically review existing literature on GNSS timing spoofing detection and mitigation techniques, identifying their strengths and limitations.
2. To analyse the behaviour of various GNSS signal parameters (pseudorange, carrier phase, Doppler measurements, position coordinates, and signal quality indicators) during timing spoofing attacks using data collected from Jammertest 2024.
3. To develop and evaluate both traditional detection methods (based on validity flags) and machine learning approach (using Isolation Forest algorithm) for identifying timing spoofing attacks.
4. To propose practical implementation frameworks for integrating these detection methodologies into existing critical infrastructure systems.

1.4 Thesis Structure

The thesis is divided into four sections, literature review, methodology, analysis and results, and finally discussion. Chapter two discusses literature review and highlights the importance of GNSS timing and recent advances in detecting and mitigating spoofing. Chapter Three outlines the methodology used to collect data in the Jammertest 2024. Chapters Four, Five, and Six present the results and their explanation, discussing various measurements made and how spoofing is detected using both traditional and AI-based methods. Chapter Seven provides discussion and highlights how the obtained results

could benefit and be implemented in the real world. Finally, the last chapter presents conclusions and future work, followed by the references.

2 GNSS Timing Spoofing: Fundamentals and Challenges

This chapter presents the literature review of the GNSS timing spoofing effects and ways on how to detect it.

2.1 Overview of GNSS and Timing Synchronization

Global Navigation Satellite System, commonly abbreviated as GNSS, identifies the orbital satellite networks that transmit positioning and timing data from space to GNSS receivers. These receiving units subsequently analyse the incoming signals to establish geographical coordinates through multi-lateration procedures (Kaplan & Hegarty, 2006, p. 3). GNSS encompasses various satellite navigation systems developed by different countries. These include the United States' GPS (NCO, 2021), the European Union's Galileo (EUSPA, n.d.), Russia's GLONASS (IAC, n.d.), India’s NavIC (ISRO, 2023), China's BeiDou (State Council Information Office of the People's Republic of China, 2016), and Japan's QZSS (Cabinet Office, 2025).

The essential mechanism of GNSS employs one-way time of arrival (TOA) ranging, in which satellites transmit ranging codes and navigation information via specific frequencies using code division multiple access (CDMA) strategies (Kaplan & Hegarty, 2006, p. 3). Each satellite generates unique codes that allow receivers to distinguish between signals from different satellites. The navigation data enables receivers to determine satellite locations at transmission time, while ranging codes help calculate signal transit time and thereby determine satellite-to-user range.

To calculate a three-dimensional position, a receiver needs signals from at least four satellites as illustrated in Figure 1. This requirement stems from the need to solve for four unknowns: latitude, longitude, altitude, and receiver clock offset from system time (Kaplan & Hegarty, 2006, p. 3). If the satellite time is known or (and) the altitude, fewer satellites are needed.

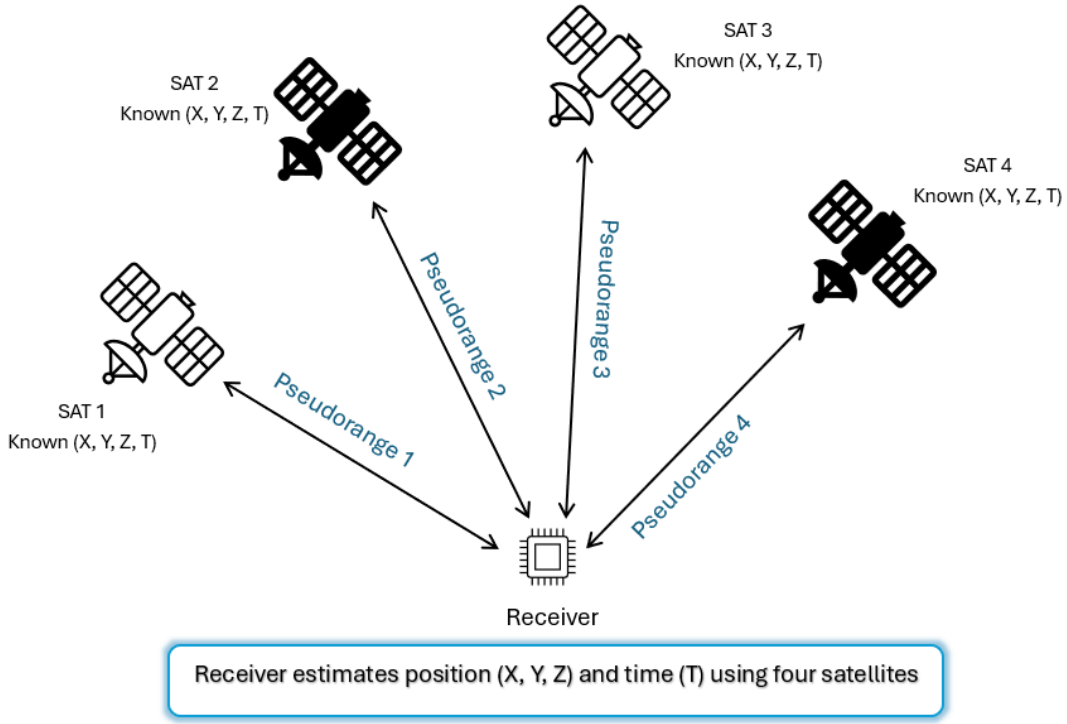

Figure 1: Position estimation using four satellites

Explaining each constellation, the United States pioneered GNSS technology with GPS (Global Positioning System), which was the first fully operational global navigation satellite system. GPS employs a standard configuration of 24 satellites distributed across six orbital planes, with four satellites positioned in each plane (Kaplan & Hegarty, 2006, p. 3). These satellites travel at approximately 20,200 km above Earth's surface and complete one orbit roughly every 12 hours. GPS provides dual service tiers: the Standard Positioning Service (SPS) accessible to civilian users and the Precise Positioning Service (PPS) restricted to U.S. military personnel and authorized government agencies. The SPS delivers horizontal accuracy better than 13m and vertical accuracy better than 22m at 95% confidence levels, while the PPS provides at minimum 22m horizontal and 27.7m vertical accuracy (Kaplan & Hegarty, 2006, p. 4).

GPS has undergone significant modernization efforts since its inception. The modernization program includes adding new civil signals (L2C and L5) and military signals (M-code)

to enhance accuracy, reliability, and resistance to interference (Kaplan & Hegarty, 2006, p. 5). These additional signals allow users to correct for ionospheric delays through dual-frequency measurements and increase robustness against interference.

The second constellation is Galileo. Galileo is the European Union's GNSS, designed specifically for civilian use worldwide. Galileo provides multiple service levels, including an open service (free of direct user charges), a commercial service, a safety-of-life service for safety-critical users, a public regulated service for government-authorized users, and support for search and rescue operations (Kaplan & Hegarty, 2006, p. 6) (NovAtel, n.d.-d). A key feature of Galileo's safety-of-life service is authentication of received satellite signals and integrity monitoring. This provides timely warnings to users when signals cannot be safely used according to specifications (Kaplan & Hegarty, 2006, p. 7). Galileo's planned constellation consists of 30 satellites and a full worldwide ground control segment. One of its primary goals is full compatibility with GPS, with measures taken to ensure interoperability between the two systems. These interoperability factors include signal structure, geodetic coordinate reference frame, and time reference system (Kaplan & Hegarty, 2006, p. 7). Right now, Galileo has 24 satellites in orbit (NovAtel, n.d.-d).

Russia's equivalent to GPS is GLONASS (Global Navigation Satellite System). Similar to GPS, GLONASS employs satellites in medium Earth orbit, ground control infrastructure, and user equipment. Currently, GLONASS maintains a constellation of 24 satellites (NovAtel, n.d.-c). The system's modernization efforts include GLONASS-M satellites with enhanced reliability and an additional civil signal, as well as GLONASS-K satellites that transmit all previous signals plus a third civil frequency dedicated to safety-of-life applications. GLONASS-K satellites are additionally configured to transmit integrity information and wide area differential corrections (Kaplan & Hegarty, 2006, p. 8). GLONASS, like GPS, operates as a dual-use system without direct fees for civilian users. Russia collaborates with both the European Union and the United States to ensure compatibility between

GLONASS and Galileo, and GLONASS and GPS, respectively (Kaplan & Hegarty, 2006, p. 8).

BeiDou, China's navigation system, represents a multiphase satellite navigation program delivering positioning capabilities, fleet management, and precise time dissemination to Chinese military and civilian users. Unlike GPS, Galileo, and GLONASS which utilize one-way TOA measurements, BeiDou initially implemented two-way range measurements through its Radio Determination Satellite Service (RDSS) (Kaplan & Hegarty, 2006, p. 9). In the RDSS approach, a central operations facility transmits a polling signal via satellite to users, who then respond with a signal through at least two satellites. The system measures transit time as signals circulate from operations center to satellite to user and back. This process enables user position determination, which is subsequently transmitted back to users (Kaplan & Hegarty, 2006, p. 9).

BeiDou was originally conceived to provide integrity and wide area differential corrections through a satellite-based augmentation system. As of 2021, the constellation comprises 45 operational satellites (Kaplan & Hegarty, 2006, p. 10) (NovAtel, n.d.-e).

Japan developed the Quasi-Zenith Satellite System (QZSS) to enhance GPS capabilities and provide mobile satellite communications for Japan and surrounding regions. The system primarily addresses GPS visibility challenges in urban environments and mountainous terrain, which Japan considers problematic across approximately 80% of its territory (Kaplan & Hegarty, 2006, p. 18). QZSS utilizes satellites in highly inclined, elliptical orbits ensuring at least one satellite remains near zenith (directly overhead) from Japan's perspective. This orbital configuration enhances satellite visibility in areas where buildings or topographical features might otherwise obstruct signals from lower-elevation satellites. As of 2024, QZSS operates with 7 satellites (NovAtel, n.d.-f).

NavIC (Navigation with Indian Constellation) is India's regional satellite navigation system designed to provide positioning accuracy of 10 meters. It is local to India mostly since

some of the satellites are in geostationary orbits. As of 2022 it has eight satellites (NovAtel, n.d.-g).

By integrating these systems, users can potentially access over 100 navigation satellites globally, significantly improving position accuracy, reliability, and availability compared to using any single system alone. All the current constellations are depicted in Figure 2. Furthermore, Global GNSS systems operate across diverse L-band frequencies with distinct signal structures. GPS, GLONASS, Galileo, BeiDou, QZSS, and NavIC each utilize different frequency allocations and coding schemes. GPS operates at 1575.42 MHz (L1) and 1227.6 MHz (L2), GLONASS using Frequency Division Multiple Access (FDMA) from 1602.0 MHz, Galileo spanning 1176.45-1575.42 MHz across four bands, and BeiDou at 1207.14-1575.42 MHz, creating a complementary spectrum coverage environment that enhances positioning accuracy and service reliability worldwide (European Space Agency, 2011).

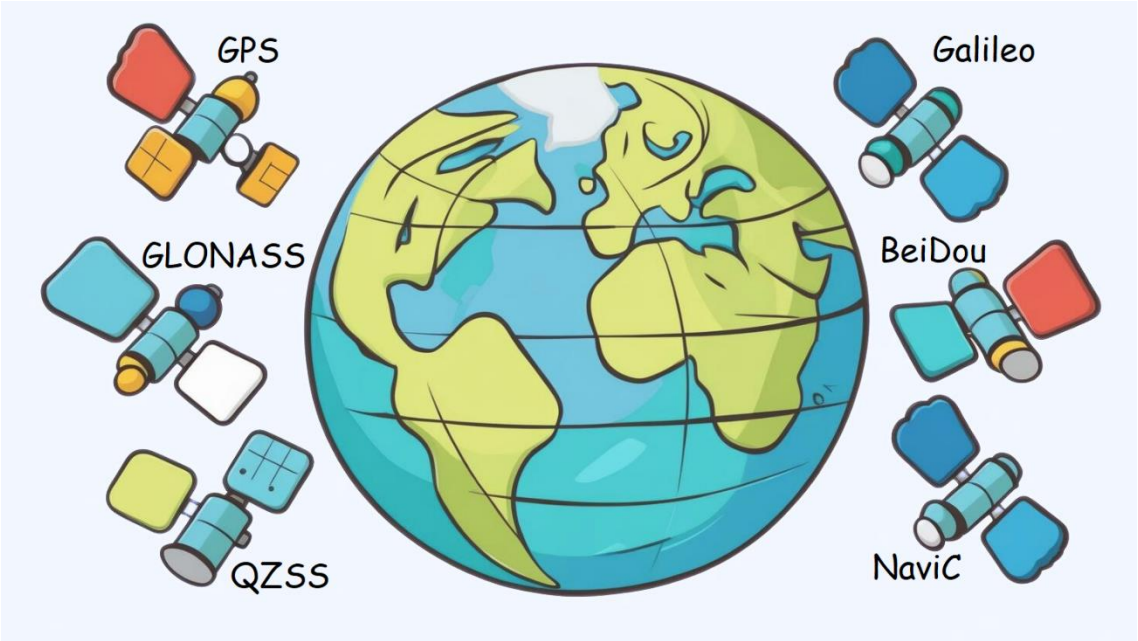

Figure 2: Current GNSS constellations

GNSS timing is one of the fundamental aspects of global navigation, providing precise temporal synchronization that enables accurate positioning services worldwide. The system architecture revolves around highly sophisticated atomic clocks installed on orbiting satellites that maintain extremely precise time measurements (European Space Agency [ESA], n.d.). The first atomic clock was developed in 1955 as shown in Figure 3 (NIST, n.d.). These timing systems employ three primary types of atomic frequency references: rubidium vapor cells, caesium atomic beams, and hydrogen masers, all functioning through quantum transitions at microwave frequencies (Hollberg, 2021). The fundamental operation relies on the consistent behaviour of atoms transitioning between energy states, which produces stable frequency outputs that serve as timing references.

The atomic clocks aboard GNSS satellites achieve remarkable stability, with rubidium clocks losing approximately three seconds per million years and hydrogen masers losing only one second per three million years (ESA, n.d.). This precision enables GNSS satellites to broadcast time signals synchronized to Coordinated Universal Time (UTC), which they receive from ground control stations (Chandler, 2022). When these signals reach Earth-based receivers, they achieve synchronization with uncertainties as low as 5 nanoseconds, effectively transferring atomic clock precision to users without requiring them to operate such sophisticated timekeeping equipment (NCO, 2022).

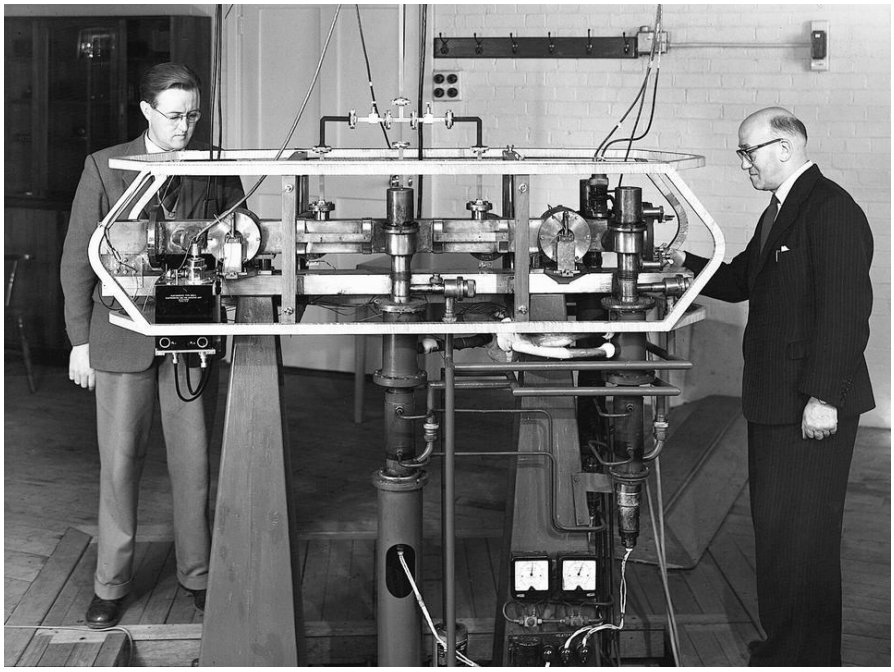

Figure 3: First caesium atomic clock (NIST, n.d.)

GNSS timing synchronization works by receiving precisely timed signals from multiple satellites with onboard atomic clocks. GPS satellites transmit radio signals at 1575 MHz that receivers capture, calculate propagation delays based on known satellite positions, and adjust for atmospheric effects. This provides extremely accurate time references (typically within 50 nanoseconds) that power systems and other applications use for synchronization across wide geographic areas (Behrendt & Fodero, 2006). GNSS timing synchronization operates through a precise one pulse per second (1PPS) signal output from GNSS receivers. Wu et al. (2016) explains that 1PPS signals synchronize devices to UTC or GNSS system time by generating pulses using a Numeric Controlled Oscillator (NCO) and adjusting them based on time differences with GNSS signals. However, timing errors occur in "sawtooth" patterns due to discrete phase calibration steps with uncorrected frequency errors. The researchers propose a zoom technique using programmable delay line technology (electronic circuits that add precise, adjustable time delays to incoming signals) with 0.25ns precision to reduce this sawtooth error, improving timing accuracy from approximately 52ns to just 1-2ns peak-to-peak.

The timing mechanism operates through a continuous broadcast of precise time signals from satellites that propagate at light speed. The receivers decode the embedded timing information, calculating differences between transmission and reception times (NASA, 2019). Some advanced timing receivers also support additional synchronization protocols like IRIG-B, which delivers time signals in binary coded decimal format once per second (Behrendt & Fodero, 2006).

The biggest application of GNSS timing is in the power grids. Time synchronization is a foundational requirement for modern power systems, enabling coordinated operation across geographically dispersed infrastructure components. Within power grids, precise timing allows for synchronized measurements, accurate event sequencing, and coordinated control actions (Zhang et al., 2020). GNSS, particularly GPS, has become the primary means of achieving this synchronization due to its ability to provide timing accuracy within nanoseconds.

According to Falletti et al. (2019), GNSS-based timing can achieve performance equivalent to atomic clocks but at significantly lower cost, making it an attractive solution for critical infrastructures. In power systems, GNSS receivers provide standardized timing outputs including 1-PPS signals, NMEA 0183 time telegrams, and IRIG-B signals that serve as common time references across the grid. These timing references synchronize various system components including phasor measurement units (PMUs), protective relays, and metering equipment.

The importance of precise timing is particularly evident in applications like phasor measurement, where IEEE C37.118 standards specify maximum acceptable synchronization errors of 31 µs for 50 Hz systems and 26 µs for 60 Hz systems (Falletti et al., 2019). In Advanced Metering Infrastructure (AMI), timing synchronization ensures accurate timestamping of electricity usage data, which is crucial for implementing time-of-use tariffs and proper billing (Bin et al., 2020). Time synchronization in power systems typically follows either hierarchical or distributed architectures. In hierarchical systems like

AMI, field devices synchronize to a central timing source at the head end, while in distributed systems like Wide-Area Measurement Systems (WAMS), each substation maintains its own timing source synchronized to GNSS (Zhang et al., 2020). Both approaches rely heavily on GNSS as the ultimate reference, establishing a critical dependency that, despite delivering superior timing performance, simultaneously introduces systemic vulnerabilities susceptible to exploitation through spoofing attacks.

GNSS time synchronization also plays a critical role in telecommunications systems, particularly in ensuring robust and precise timing for network operations. High precision timing is required in 5G networks where nanosecond-level synchronization is required for technologies like MIMO (Multiple Input-Multiple Output) and transmit diversity (Cao et al., 2024). GNSS-based time synchronization offers exceptional accuracy, with experiments showing that timing accuracy can reach ±2 microseconds for individual nodes and sub-10 microseconds among multiple nodes (Hasan et al., 2023). For telecom networks requiring continuous stability, integrating caesium atomic clocks with GNSS receivers provides enhanced robustness against signal outages. This combination ensures that even during satellite signal disruptions, time synchronization can be maintained at high precision (within 50ns) for extended periods, significantly improving network reliability for critical telecommunications infrastructure (Cao et al., 2024; Ruiqiong et al., 2019).

Apart from telecom and grid, GNSS timing plays a crucial role in the financial industry, where precise transaction timestamping is essential for market integrity and operational compliance. Major stock exchanges like the New York Stock Exchange, which processes approximately around $2 billion in trades just in the first two minutes after opening (Inside GNSS, 2014), rely on precise GNSS-synchronized clocks. The European Union's Markets in Financial Instruments Directive II (MiFID II) mandates synchronized clocks across all trading venues to ensure transaction transparency and fraud prevention (Inside GNSS, 2014). In high-frequency trading, where microseconds can determine profitability, many trading firms place GNSS receivers directly on their server room roofs to gain timing advantages (Finance Derivative, 2024). The Madrid Stock Exchange exemplifies this reliance

on GNSS, using sophisticated time services with atomic clocks synchronized to UTC via GNSS time-transfer (Inside GNSS, 2019). However, this dependence creates vulnerabilities, as spoofing attacks could potentially manipulate market timing, creating opportunities for fraudulent transactions or market disruptions similar to the 2010 "flash crash" (Quartz, 2017). This is why sophisticated methods are needed to make sure there are no GNSS interferences when it comes to timing.

2.2 Spoofing Techniques and their Impacts

When it comes to GNSS spoofing, there are several techniques used to produce false signals, particularly for timing spoofing. According to Meng et al. (2022), GNSS spoofing can be classified into three main categories based on signal generation: production spoofing, forwarding spoofing, and gradual self-synchronization spoofing. Production spoofing directly transmits signals generated by signal generation equipment to deceive the receiver, allowing attackers to manipulate transmission time and location information. Forwarding spoofing collects real satellite signals, enhances them, and delays forwarding them to cause incorrect navigation positioning. For timing attacks specifically, the spoofer can manipulate the timestamp carried in the navigation message, affecting the receiver's clock synchronization. The most sophisticated approach is gradual self-synchronization spoofing, which deceives the receiver tracking loop by gradually modifying range delay and Doppler modulation according to the target receiver's dynamic performance, enabling covert control of satellite delay timing.

When it comes to GNSS timing spoofing, Gao and Li (2022) propose three distinct algorithms that manipulate receivers in different ways. The first approach modifies pseudorange measurements while maintaining spatial position, creating a timing offset without position changes. The second algorithm alters satellite positions through navigation message parameter modifications, making it harder for receivers to detect spoofing since no pseudorange delay is added. The third combines both techniques, modifying both pseudorange and satellite positions simultaneously. Their experiments revealed

that the pseudorange modification method achieved nearly perfect timing spoofing with minimal position change, while the satellite position modification technique showed slightly reduced effectiveness but offered better concealment against detection methods that monitor pseudorange delays. Furthermore, Radoš et al. (2024) describe three main spoofing attack types: simplistic (using GNSS signal simulators to create fake signals), intermediate (receiver-based attacks that monitor authentic signals before generating synchronized fake ones), and sophisticated (using multiple transmitters from different angles). Simplistic attacks typically begin with jamming to force receivers to lock onto fake signals as illustrated in Figure 4, while intermediate attacks achieve more covert position manipulation without triggering warnings.

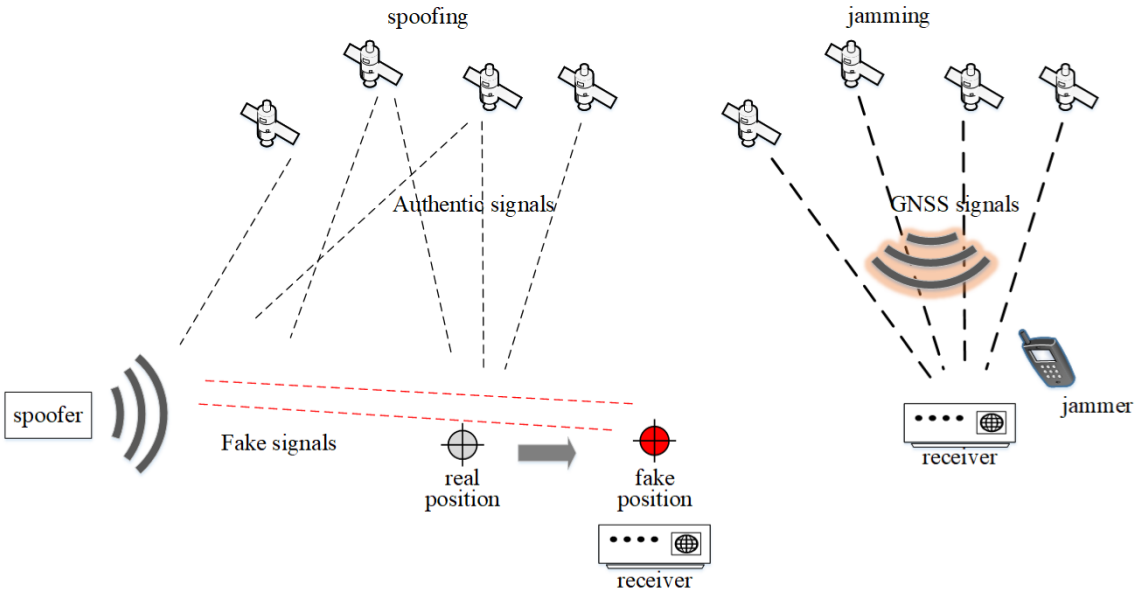

Figure 4: GNSS Spoofing and Jamming (Radoš et al., 2024)

GPS timing spoofing attacks can have significant consequences for cyber-physical systems that rely on precise timing. Wei and Sikdar (2019) demonstrated that different spoofing techniques produce varying degrees of impact. When attackers manipulate only the GPS timestamp or introduce identical delays across all satellite signals, the resulting location and pseudorange errors can be extremely small (under 248.6m for location error), making these attacks particularly difficult to detect. These subtle attacks can still cause substantial timing errors exceeding 36.5µs, sufficient to violate the IEEE

C37.118 standard for power grid operations. Such violations could potentially trigger serious disruptions in critical infrastructure including power grids, financial exchanges, telecommunications networks, and banking systems where precise time synchronization is essential.

Zhang et al. (2020) further elaborate that GPS spoofing-based time synchronization attacks (TSA) can severely compromise both hierarchical and distributed time synchronization systems in power grids. By manipulating GPS signals, attackers can disrupt critical monitoring and control systems including Advanced Metering Infrastructure (AMI), Wide-Area Measurement System (WAMS), and Lightning Location System (LLS). For instance, a time deviation exceeding 5 minutes in AMI can invalidate metering data and control commands, essentially disabling both monitoring and remote-control capabilities. In WAMS, even millisecond-level timing errors can produce significant phase angle measurement errors, potentially leading to catastrophic blackouts through incorrect state estimation.

Falletti et al. (2019) conducted experimental testing on three commercial GNSS timing receivers, revealing concerning vulnerabilities to various spoofing attacks. According to their findings, the tested receivers demonstrated vulnerability to signal spoofing, with minimal evidence indicating fake signal detection. They emphasize that timing errors in critical infrastructures like power grids could have severe consequences, particularly for synchronization-dependent applications such as phasor measurement units (PMUs), where timing errors as small as 26.5µs can again exceed IEEE C37.118 standards and potentially cause system instability.

Bin et al. (2020) specifically analyzed how TSA affects AMI systems, noting that beyond operational disruption, sustained GPS spoofing attacks could cause significant revenue losses for power utilities. Their simulation demonstrated that time deviations in smart meters could result in incorrect time-of-use tariff application, with potential daily revenue losses of over $50,000 for a utility with 44,096 MW average power. Their research

introduced a precision-enhanced oven-controlled crystal oscillator incorporating cumulative error compensation mechanisms to identify time jitter and sustain precise timing functionality during extended spoofing incidents. Apart from power grids, GNSS timing synchronization is also needed for unmanned vehicle swarms, as precise timing enables coordinated movement, formation maintenance, and distributed decision-making across multiple units. When compromised through spoofing attacks, timing errors can disrupt swarm coordination, potentially causing incomplete coverage of surveillance areas or dangerous collision scenarios (Ranganathan et al., 2023).

These collective studies underscore the necessity for robust, multi-layered defense strategies against GPS spoofing attacks, combining both signal-level detection methods and system-level analytical approaches to protect critical power infrastructure from increasingly sophisticated timing attacks.

2.3 Existing Spoofing Detection Strategies

GNSS spoofing can be detected by two major approaches, either by using traditional approach which involves signal processing or by using artificial intelligence which involves machine or deep learning. This section will highlight both and recent advancements made in them.

2.3.1 Traditional Approaches

There are many traditional approaches to detecting spoofing in GNSS. These can be classified into three major categories: signal processing methods, geometric analysis methods, and positioning methods (Radoš et al., 2024) (Psiaki & Humphreys, 2021). These categories are then further subdivided into other methods as shown in Figure 5.

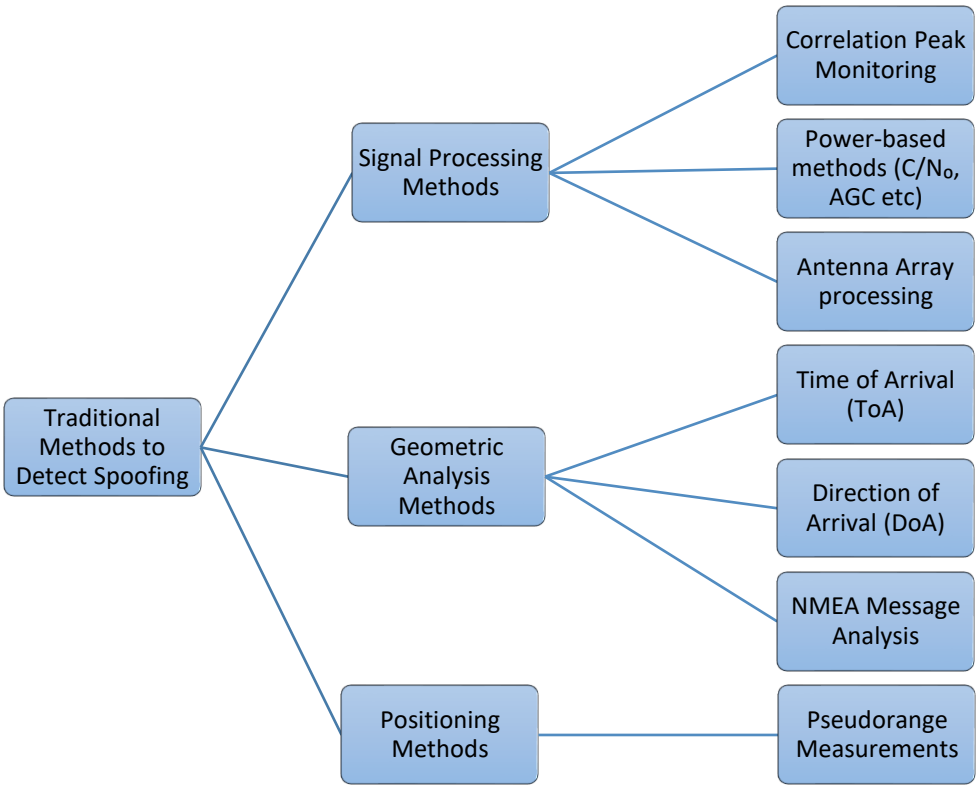

Figure 5: Traditional methods to detect spoofing.

Firstly, explaining the signal processing methods, that work by processing the signal received by the receiver and performing various analysis on it in order to detect spoofing. In correlation peak monitoring, the signal quality is monitored by observing the distribution and characteristics of correlation peaks. When a spoofing attack occurs, multiple correlation peaks may appear, or distortions in the peak shape might be detected. Li et al. (2020) proposed a method using the k-nearest neighbour (KNN) algorithm to detect spoofing signals with small delays during the acquisition phase. Their approach significantly improves detection capabilities, effectively identifying spoofing signals with delays as small as 0.6 chips with high accuracy. Moving forward, research by Yang et al. (2022) introduces a spoofing countermeasure utilizing Spoofing Correlation Peak Cancellation (SCPC). Their methodology involves extracting spoofing signal characteristics from baseband samples and generating counteractive cancellation sequences. Evaluations using the Texas Spoofing Test Battery(TEXBAT) dataset demonstrated SCPC's effectiveness in rectifying compromised navigation signals through analysis of correlation

peaks, carrier-to-noise ratio, and peak trajectory patterns. In related work, Wang et al. (2023) engineered an improved spoofing detection system that examines irregular energy distributions within quadrature (Q) channel correlators. Their technique employs noise floor estimation as a normalization reference and exhibits enhanced capabilities compared to alternative signal quality monitoring approaches, particularly during overpowered and dynamic spoofing scenarios, as confirmed through TEXBAT dataset validation.

In power-based methods, the signal power, automatic gain control (AGC), and the carrier-to-noise ratio are monitored to check for disturbances that could be caused by a potential spoofed signal. A spoofed signal typically has much higher power than the satellite signal in order to overcome the authentic signals (Radoš et al., 2024). Moreover, the spoofed signals also have a relatively constant Doppler shift due to the spoofer being mounted in a fixed location. The carrier-to-noise ratio (C/N0) is a fundamental metric used in GNSS receivers to evaluate signal quality. Rustamov et al. (2020) analysed the vulnerability of consumer devices to spoofing attacks by examining C/N0 patterns. When spoofing occurs, C/N0 values often show abnormal behaviour compared to authentic signal patterns, as an example is shown in Figure 6. Psiaki and Humphreys (2016) describe sophisticated spoofing techniques where attackers can gradually increase the power of fake signals until they capture the victim receiver's tracking loops, then "drag" the receiver to false position coordinates. These techniques avoid the need for jamming and reacquisition, making them harder to detect. To counter such threats, monitoring systems can establish baseline C/N0 values for normal operation and flag significant deviations that might indicate spoofing attempts. When it comes to timing specifically, Qian et al. (2020) detects GPS timing spoofing in advanced metering infrastructure by implementing a time jitter detection system that identifies abnormal satellite clock behaviour. Their methodology uses an advanced oven-controlled crystal oscillator featuring cumulative error correction to detect timing inconsistencies. When timing jitter exceeds a 0.2μs threshold, the system automatically switches to the compensated local timing source, maintaining accurate synchronization during attacks while preserving metering

operations. Honkala et al. (2020) demonstrate that Automatic Gain Control (AGC) provides an effective mechanism for detecting both jamming and spoofing attacks on GNSS timing receivers by monitoring significant deviations in AGC levels. Their experiments show AGC responds more reliably than carrier-to-noise ratio measurements, providing earlier detection of timing threats.

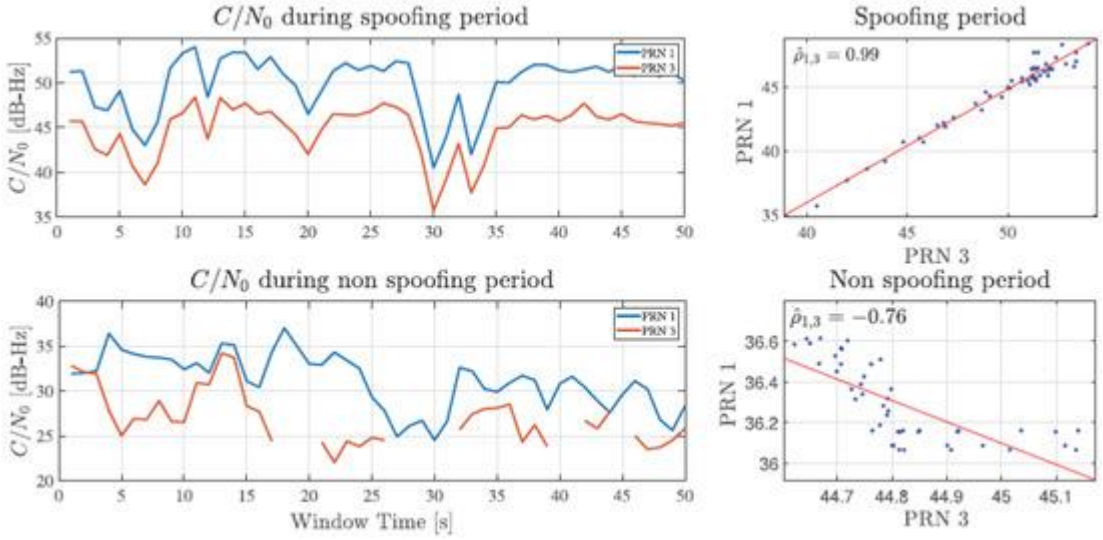

Figure 6: C/N0 during spoofing and non-spoofing windows (Radoš et al., 2024).

In antenna array processing, multiple antennas are used to detect spoofing by analysing spatial characteristics of incoming signals. This approach leverages the fact that authentic GNSS signals come from different satellites across the sky, while spoofed signals typically originate from a single source (Radoš et al., 2024). Chen et al. (2024) proposed an innovative approach using three low-cost collinear antennas to detect spoofing. Their method leverages the collinearity information to improve pointing vector estimation accuracy and employs a binary statistical detection model for real-time spoofing detection. Remarkably, their system achieved 100% spoofing detection accuracy with just a 1-meter baseline, while reducing the standard deviation of pointing vector angle deviation by over 55% when spoofing signals were present. Yang et al. (2023) developed a six-array spoofing-interference-monitoring antenna system that combines peak monitoring with an airspace-trapping algorithm. Their approach uses long- and short-baseline algorithms to quickly search the entire circumferential ambiguity, achieving directional accuracy

within 2° for spoofing signals in outdoor experiments. For mobile applications, Liu et al. (2023) introduced a method using a moving array antenna for locating spoofing sources. Their technique first extracts spoofing signals using double-differenced carrier phase characteristics, then fuses carrier phase single-difference data from multiple observation points as the antenna moves to directly localize the spoofing source. This approach avoids the data correlation issues found in traditional two-step direction of arrival estimation methods, providing both robust performance and high accuracy.

The second major category, geometric analysis, can be further divided into three key approaches: Time of Arrival (ToA), Direction of Arrival (DoA), and NMEA message analysis. In ToA approaches, the propagation time of signals is analysed to detect spoofing. Zhang and Zhan (2018) proposed a low-cost spoofing detection system based on Time Difference of Arrival Estimation (TDOAE) using two standard receivers. Their system leverages a fundamental principle: authentic satellite signals originate from various directions in space, resulting in diverse TDOAE values, while spoofed signals come from a single source, producing nearly identical TDOAE measurements. Through careful mathematical formulation and hypothesis testing, their system achieved an impressive 99.99% detection rate with less than 0.001% false alarms in both simulations and real-world tests.

DoA techniques analyse the angular information of incoming signals. Mao et al. (2023) developed a cost-effective spoofing detection system using commercial GNSS components. Their innovation addresses common challenges with low-cost antennas, including phase center instability and sampling time inconsistencies across multiple receivers. The prototype achieved 100% detection rate in open environments with 5° directional accuracy. Their technique is shown in Figure 7. For systems with a single antenna, Chen et al. (2024) developed a novel technique leveraging the angular intersection of two arrival directions (IA-DoA) using a rotational antenna setup. Their method compares estimated and predicted IA-DoA values to identify inconsistencies that indicate spoofing, particularly effective against multi-agent spoofing attacks. Similarly, Chang et al. (2022)

proposed a rotating single-antenna detection method based on an improved probabilistic neural network (IPNN). Their approach analyses how carrier-phase double differences change with satellite incident angles during antenna rotation, achieving 98.84% spoofing detection accuracy. For enhanced localization of spoofing sources, Xie et al. (2022) developed a dispreading direct position determination (DS-DPD) algorithm that leverages prior knowledge of satellite code sequences to improve accuracy by more than tenfold, even with low interference-to-noise ratios.

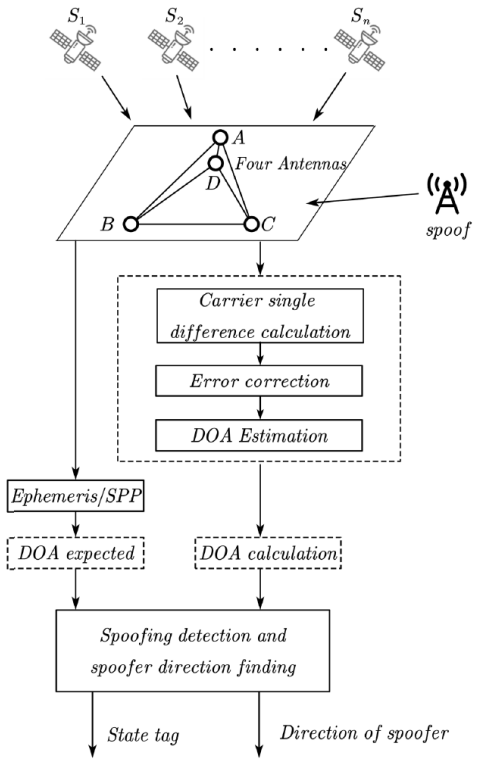

Figure 7: Spoofing detection using four antennas (Mao et al., 2023)

The NMEA message analysis approach examines standard navigation message data for anomalies. Spravil et al. (2023) developed the NMEA-based Anomaly detection (MANA) framework, which monitors NMEA-0183 sentences from maritime GPS receivers to detect spoofing without requiring hardware modifications. Their system combines multiple software-based detection methods to identify inconsistencies and integrity violations in navigation data streams that indicate potential spoofing attacks. Lee et al. (2020) demonstrates how NMEA messages can detect timing spoofing by analysing message

irregularities. When spoofing occurs, discrepancies emerge in GGA, RMC, and VTG (all different NMEA message formats. GGA has time, position and DOP (dilution of precision) etc, RMC also has same with addition of velocity except DOP and VTG has velocity, heading etc) timing data compared to expected values, allowing detection without complex raw measurements processing, making it suitable for legacy receivers with limited computational resources. In this thesis, a similar NMEA message analysis method is used to detect spoofing from the jammertest data since u-blox has a built-in validity flag that tells if the NMEA message is authentic or not. Actual details on how the receiver determines if it is spoofed are classified since its their trade secret. More details are provided in chapter five.

Lastly, considering from the positioning methods, pseudorange measurements provide another approach to spoofing detection. Xiao et al. (2019) developed a method using pseudo-range double-differences (PRDD) between two receivers. Their technique analyses discrepancies between actual PRDD measurements and expected PRDD estimations to identify spoofing. The system accounts for unknown receiver attitudes and creates a statistical decision variable for detection. Monte Carlo simulations (computational algorithms that use random sampling to estimate mathematical or physical systems' outcomes and probabilities) showed impressive results, achieving 99.99% detection probability with only a 0.001 false alarm rate using a 10-meter baseline. Another approach is looking at the pseudorange rms errors, which is also used in this thesis to identify spoofing. This technique will be explored in greater detail in chapter five. Another study detects GNSS timing spoofing by analysing clock bias change covariance between satellite pairs. By exploiting the synchronized variation patterns in spoofed signals, Jia and Liao (2025) developed an algorithm that detects Time Synchronization Attacks with 17.86% faster response time than previous methods. This computationally efficient approach requires minimal processing resources while maintaining high detection sensitivity, making it ideal for resource-constrained devices that require protection against sophisticated timing attacks. Furthermore, another study by Gao et al. (2023) presents a novel method for detecting GNSS time synchronization attacks by analysing the synchronicity patterns

in pseudorange measurements. Their Separate Clock Drift Matched Filter (SCD-MF) calculates individual clock drifts from different satellite signals and monitors their abnormal similarity using matched filtering techniques. This low-computational approach requires no precise clock model, making it more efficient than existing methods while detecting even subtle timing attacks with higher sensitivity.

Although most of these techniques are generic traditional approaches to detect GNSS spoofing, they can still work on timing spoofing and are not just bound to position or navigation types. However, some studies do not fall under the umbrella of either traditional or AI based approaches but are still effective in combating GNSS timing spoofing. A notable approach mitigates GNSS timing spoofing by implementing a sparse optimization framework that identifies malicious signal patterns in derivative domains. By observing timing attacks that exhibit sparsity in higher-order derivatives, the researchers developed a novel linearization method that jointly estimates authentic PVT states while identifying spoofing components (Lee et al., 2023). This approach formulates a convex quadratic program that effectively distinguishes between authentic signals and spoofing attacks, successfully reducing timing errors in both stationary and low-dynamic receivers without requiring additional hardware modifications or cryptographic techniques.

In conclusion, traditional GNSS spoofing detection methods encompass signal processing, geometry processing, and positioning approaches. These techniques analyse correlation peaks, signal power, antenna arrays, arrival measurements, and pseudorange data to identify inconsistencies that reveal spoofing attempts, providing essential protection for critical timing applications.

2.3.2 AI Based Approaches

Artificial intelligence is transforming satellite communication by addressing complex challenges like beam hopping, interference detection, and channel modelling (Fourati & Alouini, 2021). In terms of spoofing detection and mitigation, extensive work has been done with promising results. Like traditional approaches, AI-based methods can be

classified into three major categories: Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL). Both ML and DL can be further divided into supervised and unsupervised approaches as shown in Figure 8.

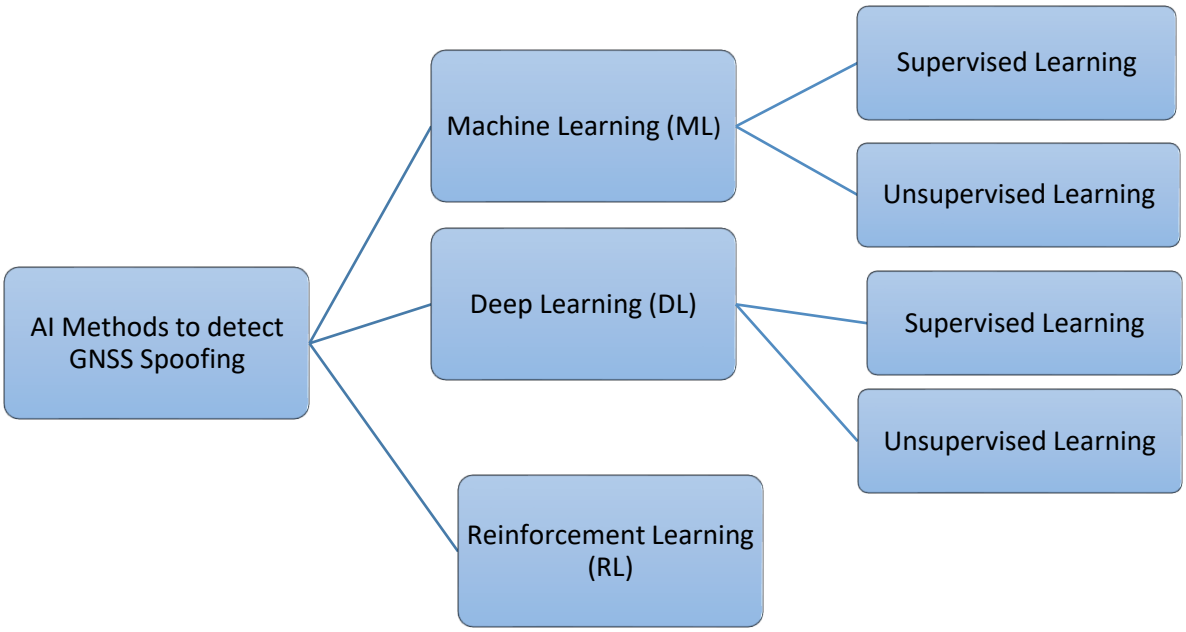

Figure 8: AI methods to detect GNSS spoofing

Referring to the flowchart in Figure 8, the ML approach is explored first. A study by Khoei et al. (2022) compares various machine learning approaches for detecting GPS spoofing attacks on UAVs. Their research evaluates nine different models, including both supervised models (Gaussian Naïve Bayes, CART, Random Forest, L-SVM, LR, ANN) and unsupervised models (PCA, K-means, Autoencoder). Through comprehensive performance analysis using metrics like accuracy, detection probability, and processing efficiency, they found that the Classification and Regression Decision Tree (CART) model outperforms others in effectively detecting GPS spoofing attacks on UAVs. Figure 9 shows the performance of various models.

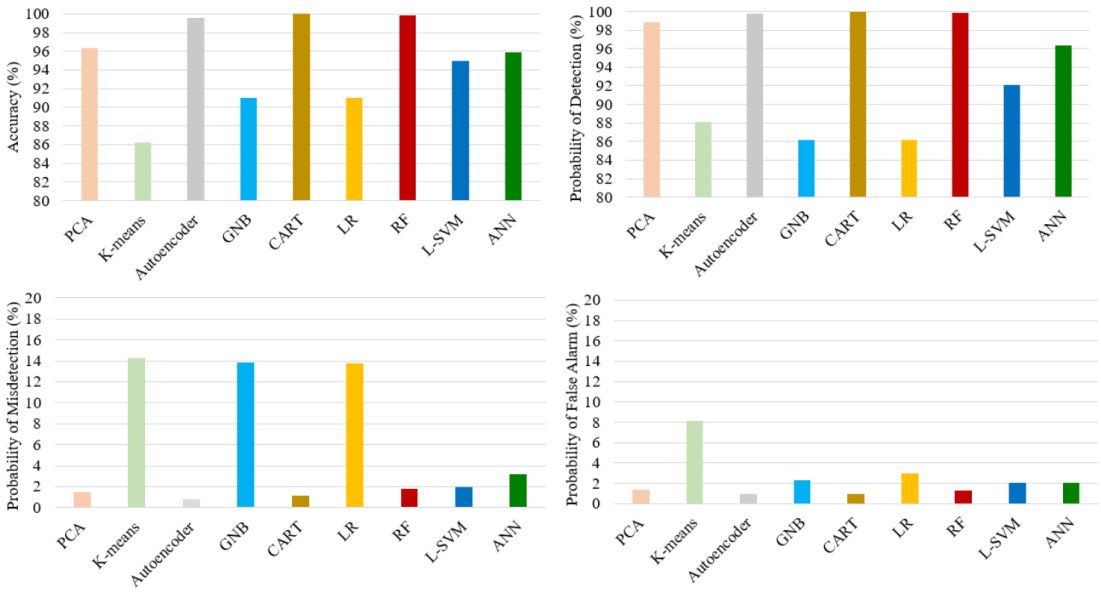

Figure 9: Performance of difference ML models in detecting GPS spoofing (Khoei et al., 2022).

Shafique et al. (2021) developed a machine learning approach to detect GPS spoofing attacks on UAVs by analysing signal characteristics. Their method utilizes Support Vector Machine (SVM) with polynomial kernel combined with K-fold analysis and voting techniques (hard and soft voting). The system classifies signals using features like jitter, shimmer, and frequency modulation. Their model achieved 99% accuracy in distinguishing between authentic and spoofed GPS signals, significantly outperforming previous detection systems. When it comes to timing specifically, Wei et al. (2022) developed an innovative GPS spoofing detection approach that exploits the statistical correlation between consecutive GPS signals. Their method analyses the Power Spectral Density (PSD) of received GPS signals using a windowed approach, then applies a statistical runs test to quantify signal correlation. This correlation data trains supervised learning algorithms, particularly CART, to identify timing attacks. Their experimental results demonstrate detection rates exceeding 95% with minimal false alarms, effectively protecting Phasor Measurement Units (PMUs) from malicious time desynchronization in power grids. Just like the previous study by Khoei et al. (2022), CART is proven to be better in detecting spoofing. In another study, Iqbal et al. (2023-a) developed a machine learning approach

that detects GPS spoofing attacks targeting Phasor Measurement Units without waiting for position-velocity-time solutions. They used Random Forest Classifier, Support Vector Machines, K-Nearest Neighbours, Gradient Boost, and Artificial Neural Network algorithms**.** Their framework extracts seven complementary features from radio frequency and tracking stages of GPS receivers, including received power and signal quality metrics. Testing five different classifiers on the TEXBAT dataset demonstrated over 99% detection accuracy with minimal false alarms, providing early warning capabilities that protect power grid synchronization integrity. While their previous work used traditional machine learning approaches, the same authors introduce a novel representation learning technique using Variational Autoencoders (VAE) in a different study. Iqbal et al. (2023-b) demonstrate that their VAE-based method can detect GPS timing spoofing attacks on synchrophasors by learning only from authentic signal patterns. This unsupervised approach outperforms supervised methods, especially on subtle attacks like DS-7, achieving 98% detection probability with only 2.5% false alarms, without requiring examples of all possible attack scenarios during training. Another study by Shereen et al. (2022) uses Graph Signal Processing (GSP) to model power grid structures as graphs, allowing detection of PMU time synchronization attacks that traditional methods miss. Their approach combines GSP with machine learning algorithms to identify attacks with high accuracy, even those specifically designed to be undetectable by conventional methods.

Now coming to deep learning approach, extensive work has been done to mitigate GNSS spoofing, especially the timing one. Romaniuc et al. (2024) implemented a novel Long Short-Term Memory (LSTM) neural network to detect NTP spoofing attacks affecting GNSS-synchronized time servers. Their approach monitors key timing parameters which include Modified Julian Date, Clock Offset, Roundtrip Delay, Dispersion, and RMS Jitter, and analyses their statistical patterns to identify anomalies. When tested against a simulated attack where timestamps were manipulated by 16 years, their LSTM algorithm successfully detected the attack by recognizing reconstruction errors between expected and actual timing values. Li et al. (2025) developed a real-time GNSS time spoofing detection framework that processes multi-satellite feature data using correlation

coefficient screening and local standardization for efficient computation. They implemented AdaBoost, Random Forest, BP neural network, and SVM machine learning models with their framework, achieving F1 scores above 99% and reducing computation time by tenfold compared to traditional methods, with response times under 10 μs. Huang and Li (2022) developed a neural network approach for detecting GPS time synchronization attacks against PMUs by implementing "phase coding" to capture relationships between amplitude and phase angles in phasor measurements. Their vector neural network with dynamic routing learns encoded relationship vectors, achieving over 95% detection accuracy across various IEEE bus systems while detecting multiple simultaneous attacks across up to five buses with better performance than traditional likelihood-based methods.

Lastly, coming to reinforcement learning approaches in GNSS spoofing detection, researchers have explored how autonomous learning agents can identify deception patterns without explicit programming. Ma et al. (2024) developed a novel deep reinforcement learning approach for UAV GPS spoofing that doesn't require prior knowledge of victim UAV reference trajectories or internal Kalman filtering parameters. Their Deep Reinforcement Learning-Navigation Deception (DRL-ND) algorithm generates deceptive position estimates based solely on radar-detected UAV motion information, using Twin Delayed Deep Deterministic Policy Gradient (TD3), Deep Deterministic Policy Gradient (DDPG), and Soft Actor-Critic (SAC) methods to learn optimal deception strategies that remain below detection thresholds while successfully redirecting UAVs to false destinations. In another study, Dasgupta et al. (2022) developed a deep reinforcement learning (DRL) approach for detecting GNSS spoofing attacks in autonomous vehicles that doesn't require predetermined rules. Their method uses low-cost in-vehicle sensor data to detect sophisticated turn-by-turn spoofing attacks by comparing predicted versus calculated distance travelled, achieving 99.99-100% accuracy and 100% recall in testing. The Deep Q-Network (DQN) agent intelligently adjusts detection thresholds to maximize spoofing identification while minimizing false positives, demonstrating how reinforcement learning can effectively model complex spoofing patterns through environmental

interaction rather than relying on predefined rules. While not specific to timing spoofing, RL can effectively model the complex patterns of spoofing attacks by learning from environmental data rather than relying on predefined rules or thresholds. Unlike traditional methods that require specific knowledge of attack characteristics, RL approaches can adapt to novel spoofing techniques through continuous learning. This also presents a research gap that needs to be addressed.

2.4 Challenges in Spoofing Detecting and Mitigation

Detecting GNSS timing spoofing presents several significant challenges due to the sophisticated nature of modern spoofing techniques. Radoš et al. (2024) emphasize that the increasing availability of low-cost software-defined radios (SDRs) has made spoofing more accessible to potential attackers, while detection methods must constantly evolve to counter these threats. A fundamental challenge is distinguishing between authentic signal degradation and spoofing, as Lee et al. (2023) note that concealed spoofing attacks must maintain synchronicity between satellite channels to avoid detection, requiring monitoring of clock bias changes across multiple satellites. Similarly, Gao et al. (2023) highlights that detecting slow-changing spoofing signals requires specialized filtering techniques, as subtle timing manipulations can evade traditional threshold-based detection methods. The detection challenge is particularly acute for timing applications, as spoofing attacks targeting time synchronization can be more subtle than position spoofing yet equally damaging to critical infrastructure. Lee et al. (2020)'s research reveals a key challenge in timing spoofing detection: smartphone GNSS receivers with high sensitivity may still be vulnerable when attackers target single constellations with elevated noise levels that mask authentic signals, making detection particularly difficult when multiple constellations aren't available for cross-verification

When examining timing specifically, Wei et al. (2022) highlight that attackers can manipulate GPS timing by maliciously desynchronizing PMUs through ephemeris manipulation or signal propagation time alterations, with even small timing errors exceeding 26.5 μs

potentially causing power grid blackouts. Traditional detection methods often struggle with subtle attacks where spoofers align carrier phase with authentic signals, as demonstrated by Iqbal et al. (2023-a) with the TEXBAT DS-7 dataset where supervised ML approaches achieved only 36% detection probability.

Romaniuc et al. (2024) note that timing synchronization vulnerabilities extend beyond power grids to critical infrastructures, with NTP spoofing attacks proving difficult to distinguish from normal operations. The challenge is further complicated by what Iqbal et al. (2023-b) describe as the "zero-day attack problem," where detectors must identify previously unseen attack patterns without exhaustive training on all possible scenarios. This suggests that unsupervised approaches like representation learning may provide more robust detection capabilities for emerging spoofing threats.

A significant challenge in GNSS timing spoofing detection involves distinguishing between environmental signal degradation and actual attacks, as Ma et al. (2024) noted that spoofing often mimics natural signal behaviour. Developing reliable real-time detection systems is further complicated by computational constraints in receiver hardware (Dasgupta et al., 2022), while the diversity of spoofing techniques requires multi-modal detection approaches that monitor signal characteristics across domains (Zidan et al., 2020). Wei and Sikdar (2019) demonstrate that GNSS timing attacks can be particularly deceptive when implemented by inserting identical delays to all satellite signals, creating minimal pseudorange errors (below 5ns change in timing due to these errors) while still causing significant timing disruptions. This approach maintains consistent receiver location errors (283.6m) yet can effectively manipulate timing enough to threaten power grid operations. Such challenges necessitate adaptive algorithms, like the isolation forest approach utilized in this thesis, capable of identifying anomalous timing behaviour without extensive prior knowledge of specific attack vectors.

3 Jammertest Dataset and Experimental Setup

This chapter presents the methodology of this thesis and how the data was collected. Jammertest is an annual event held regularly in Norway with the purpose of testing various spoofing and jamming conditions. Since this thesis takes data from Jammertest 2024, this section will highlight how the data was collected and what were the conditions regarding the data collection.

3.1 Description of Jammertest Data

Jammertest is an annual event hosted in Andøya, Norway, which stands as the world's largest open test for PNT/GNSS resilience, challenging navigation, and positioning systems against real-world interference (Jammertest, 2025). It is organized in partnership with national agencies such as the Norwegian Public Roads Administration, Communications Authority, and Defence Research Establishment, among others. This unique event offers four specialized test zones namely Bleik, Starve, Grunnvatn and the airport site as shown in Figure 10. Participants face jamming, spoofing, and meaconing attacks under dynamic outdoor conditions in these sites, testing out their equipment. With its natural geography, Andøya allows for high-power signal transmission tests while limiting impacts on public infrastructure, hence the reason for having this site for Jammertest. Many articles with relevant results including both academic and industrial continue to be published with the help of data gathered during the jammertest.

Figure 10: Different sites of Jammertest (2024-b)

Many tests were conducted during the jammertest, some focusing on spoofing and some on jamming attacks. Each test was given a unique event number to identify and get the information regarding its conditions from the log report. Since the useful data for this thesis is mainly gathered from Event 2.4.2, which was a dedicated timing spoofing experiment, this thesis will primarily focus on that and explain the analysis performed on that data.

3.2 Data Collection Process and Preprocessing

The data was gathered by the researchers at the University of Vaasa using three main devices: the u-blox F9P receiver, a Samsung Galaxy A23, and a Google Pixel 6. Unfortunately, out of all the data collected, only the log from Event 2.4.2 in the Jammertest dataset was relevant for this thesis, as it was the only one associated with GNSS timing spoofing. Within that event, only the data recorded by the u-blox receiver appeared to contain timing spoofing signals. Both mobile phones were switched on after the spoofing had ended, making their data unusable for this analysis.

3.2.1 Receiver Details

The u-blox ZED-F9P-00b-02 GNSS receiver was utilized for data collection during Jammertest 2024. This device can receive GNSS signals from multiple satellite constellations including GPS, Galileo, BeiDou, and GLONASS. It features centimetre-level accuracy with multi-band RTK and integrated support for standard RTCM corrections. The module exhibits a cold start time of 24 seconds and provides position accuracy of 0.01 m. Additionally, it incorporates active CW detection and removal capabilities with an onboard bandpass filter, along with sophisticated anti-spoofing algorithms such as Galileo open service navigation message authentication (OSNMA) (u-blox, 2024).

OSNMA is a cryptographic mechanism that allows users to verify the authenticity of navigation data, protecting against spoofing attacks by ensuring signals originate from legitimate Galileo satellites rather than malicious sources. It uses the Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol to provide cryptographic authentication data within the E1 I/NAV message to perform the checks (European GNSS Service Centre, 2025). The TESLA protocol is a broadcast authentication method using symmetric cryptography and loose time synchronization to make sure data is genuine (Perrig et al., 2002). However, in our case, the OSNMA feature of u-blox receiver was switched off.

The receiver is installed on its module board which makes it easier to connect it to external antennas and a computer to gather and view the data. The module features multiple communication interfaces including UART, SPI, I2C, and USB 2.0 FS. It operates on a voltage supply of 2.7-3.6V with typical current consumption of 85mA when tracking multiple constellations. The module can simultaneously track multiple satellite constellations (GPS, GLONASS, Galileo, BeiDou) on different frequency bands including L1/L2C for GPS, L1OF/L2OF for GLONASS, E1-B/C/E5b for Galileo, and B1I/B2I for BeiDou. It

supports QZSS and various SBAS systems (WAAS, EGNOS, GAGAN, L1Sb) (u-blox, 2024). The USB and antenna are both connected on the bottom side as shown in Figure 11.

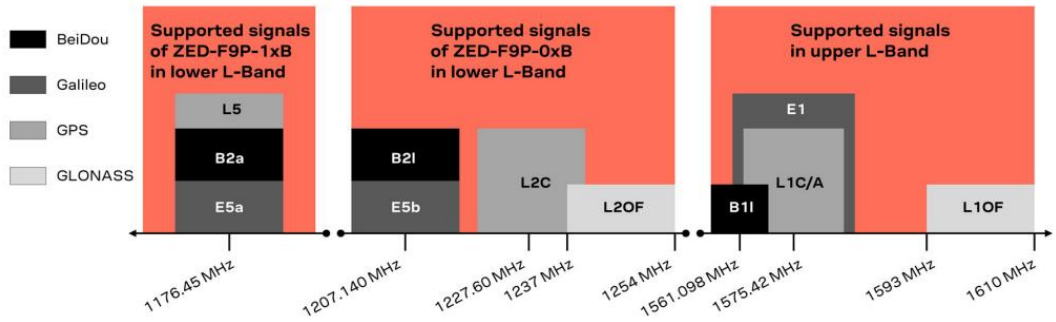

Figure 11: Frequencies of the ZED receiver (u-blox, 2024)

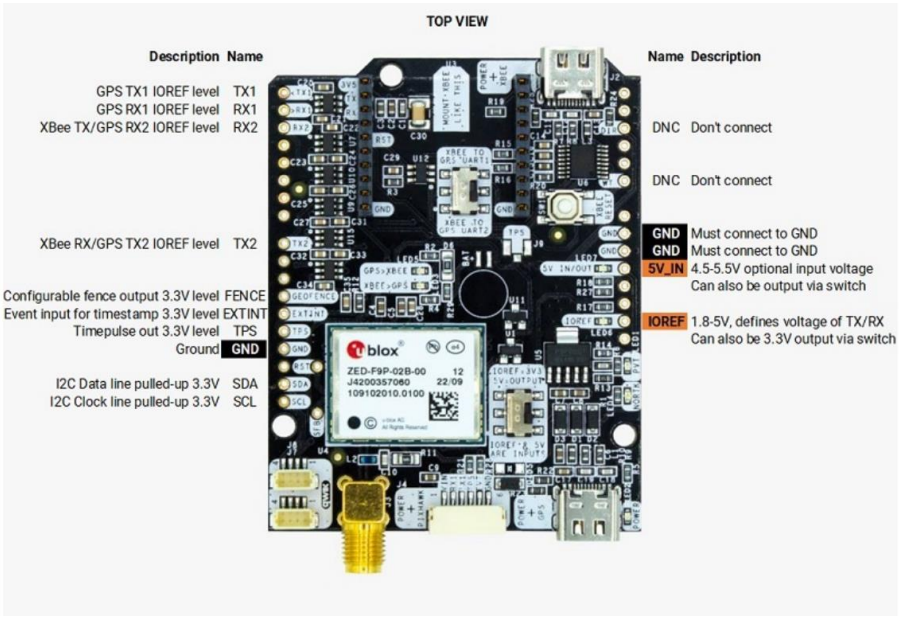

Figure 12: ZED-F9p-00b-02 with its board (ArduSimple, 2025)

The data produced by the receiver is in the ubx format. A simple and straight forward way to view the data would be using the u-center software which is the official software released by the u-blox to configure the receivers and replay their data. However, u-center lacks critical diagnosis of the data and for that reason, most of the analysis done during this thesis was completed using the pyubx2 Python library.

3.2.2 Test Details from Log

The data gathered during the time spoofing event in the ubx file spanned from 07:17 UTC to 08:45 UTC (Universal Time Coordinated) with a total duration of 1 hour, 27 minutes and 14 seconds. This data was collected in Bleik. The test logs from Jammertest 2024 provide detailed data on the tests conducted for various jamming and spoofing conditions (Jammertest, 2024-a). Since the log file is in Central European Summer Time (CEST), it is 2 hours ahead of the UTC. So 9:04 CEST corresponds to 7:04 UTC. As our data starts from 7:17, it corresponds to 9:17 entry in the log when the spoofing power was at 0 dBm.

For the event 2.4.2, the spoofing of 900 seconds into the future started at 7:04 UTC (9:04 CEST) with an initial power of -35 dBm, which increased after every two minutes with an increment of 5 dBm. It went all the way up to 30 dBm and then had a sudden drop to -15 dBm for the event 4.2.3 with time offset of 3 minutes into the past for this. However, the u-blox f9p receiver showed excellent resilience and was only affected during the event 2.4.2 (see details in chapter 4). The ubx file also contains data from the 2.4.12 and 2.4.13 events but those are irrelevant to the scope of this thesis. Details of the test conducted during the duration of the ubx file are given in Table 1.

Table 1: Jammertest event details for the ubx file (Jammertest, 2024-a).

Test ID	Test name	Date	Start CEST)	Stop (CEST)	Comment	Jamming power (W)	Jamming power (dBm)	Spoofing power (dBm)
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.04.10	9.06.10				-35
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.06.10	9.08.10				-30

2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.08.10	9.10.10				-25
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.10.10	9.12.10				-20
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.12.10	9.14.10				-15
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.14.10	9.16.10				-10
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.16.10	9.18.10				-5
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.18.10	9.20.10				0
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.20.10	9.22.10				5
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.22.10	9.24.10				10
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.24.10	9.26.10				15
2.4.2	Time offset 15 minutes from real	2024-09-12	9.26.10	9.28.10	Spoofing ramp continued			20

	time, with power ramp				higher than TP			
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.28.10	9.30.10	Spoofing ramp continued higher than TP			25
2.4.2	Time offset 15 minutes from real time, with power ramp	2024-09-12	9.30.10	09:32:27	Spoofing ramp continued higher than TP			30
2.4.3	Time offset -3 minutes from real time, with power jump	2024-09-12	9.50.10	10.00.10				-20
2.4.3	Time offset -3 minutes from real time, with power jump	2024-09-12	10.00.10	10:05:16				15
2.4.12	Static + Pseudorange error	2024-09-12	10.20.10	10:35:15	Increasing pseudorange error in the test period of 5 to 15 min, up to 1800 m. This gives a pseudorange error of 3 m/s, equivalent to a time error of 9 ns/s. A total accumulated time error of 6 µs			15
2.4.13	Static + Pseudorange error, with initial and continous jamming	2024-09-12	10.50.08	10.50.21	Initial jamming (E6, L2, E5b, L5)		35	
2.4.13	Static + Pseudorange error, with initial and continous jamming	2024-09-12	10.50.21	10.55.19	Jamming of L1, G1, B1I activated		35	
2.4.13	Static + Pseudorange error, with initial and	2024-09-12	10.55.19	10.55.23	Spoofing activated. Spoofing power different than TP		35	15

	continous jamming							
2.4.13	Static + Pseudorange error, with initial and continous jamming	2024-09-12	10.55.23	10.55.24	Jamming of E5b deactivated		35	15
2.4.13	Static + Pseudorange error, with initial and continous jamming	2024-09-12	10.55.24	10.55.25	Jamming of L5 deactivated		35	15
2.4.13	Static + Pseudorange error, with initial and continous jamming	2024-09-12	10.55.25	10.55.26	Jamming of L2 deactivated		35	15
2.4.13	Static + Pseudorange error, with initial and continous jamming	2024-09-12	10.55.26	11:05:21	Jamming of L1 deactivated. Time error of 9 ns/s. A total accumulated time error of 6 µs		35	15

3.3 Testbed Setup and Assumptions

The data was collected using the u-blox f9p receiver as mentioned in section 3.2. Initially the receiver is at a fixed position (69.27547832 in latitude and 15.96832496 in longitude) and so is the spoofer. The spoofer is located at, Latitude: 69.27547832 and Longitude: 15.96832496 with an altitude of 35 m (Jammertest, 2024-b). The distance between the spoofer and receiver can be determined using the Haversine formula, which calculates great-circle distances between two points on a sphere based on their respective longitudes and latitudes. This mathematical approach represents an approximation as it assumes Earth is spherical. In reality, Earth exists as an oblate spheroid with its radius

varying from 6,357 km at the poles to 6,378 km at the equator (Moritz, 2000). Despite this approximation, the Haversine formula is highly accurate for short distances, with an error margin of less than 0.5% (Azdy & Darnis, 2020) (Upadhyay, 2019) (Agafonkin, 2016). The formula is given in equation 1:

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta\text{lon}}{2}\right) \quad (1)$$

where, Δlat is the difference between latitudes and Δlon is the difference between longitudes. Once we have the a , we can calculate the central angle c :

$$c = 2 \cdot \arcsin\left(\sqrt{a}\right) \quad (2)$$

After obtaining the central angle, we can finally compute the distance d using:

$$d = R \cdot c \quad (3)$$

Where R is the radius of Earth (approximately 6,371,000 meters). The distance between spoofer and receiver is then computed to be approximately 34.91 meters. Figure 13 shows it on a map:

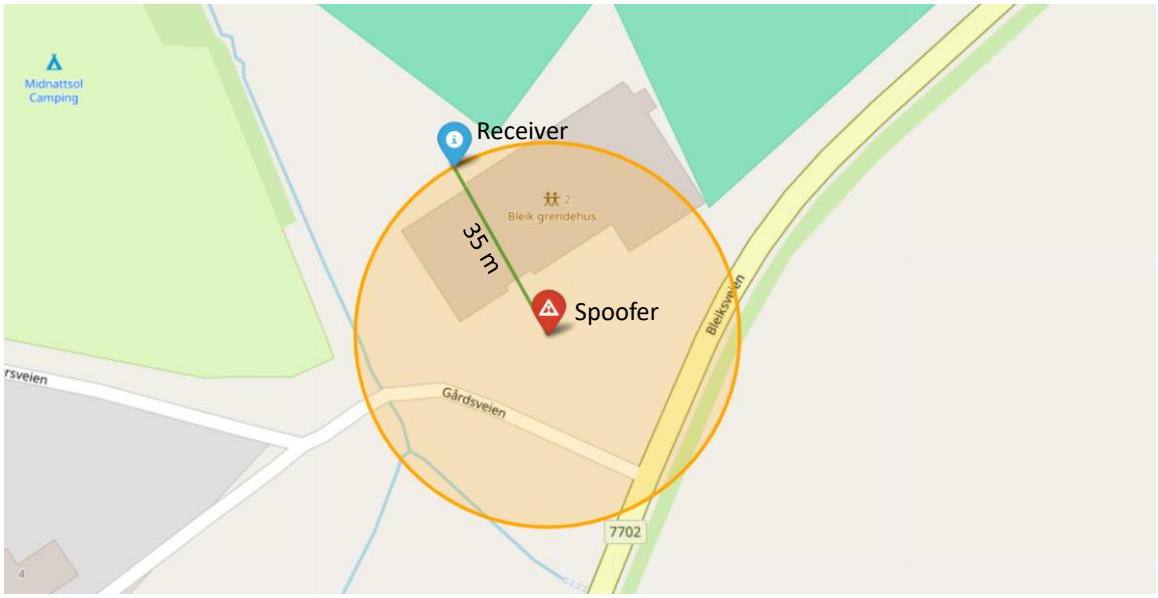

Figure 13: Distance between receiver and spoofer.

This test employed a Cigarette-type (which could be mounted in a car's cigarette port) GNSS spoofer with power ranging from 3.16e-07W to 0.0316W, targeting bands L1, L2, L5, E1, E5a, and E5b. The test aimed to evaluate equipment response to misleading GNSS-PNT information, particularly timing. The spoofed signals were not consistent with actual satellite transmissions; however, they successfully held the receiver's navigation fix at the intended location. These signals were generated from a stationary antenna using different ephemerides and spanned multiple GNSS bands and constellations. Some scenarios began with 5-minute jamming periods, while others featured continuous jamming. Tests were separated by breaks allowing receivers to get authentic satellite signals again. According to the Jammertest (2024-b), the transmitter had a range of a few hundred meters so the receiver being only 35 meters away was well within the effective range.

4 Analysis of Timing Spoofing Event in Jammertest

The analysis of the .ubx file generated during the event 2.4.2 by the Ublox receiver was conducted using Python, specifically the pyubx2 library. It was first developed around 2020 with still getting latest version updates as of April 2025 (semuadmin, 2025). The pyubx2 library is a Python 3 package designed to not only parse but also generate UBX protocol messages for u-blox GNSS/GPS devices. It supports UBX, NMEA 0183, and RTCM3 protocols, making it possible to extract and build GNSS data such as position, velocity, and time details (semuadmin, 2025).

Since most of the analysis was done in Google Colab environment, pyubx2 can be easily installed using pip command. After installation, `UBXReader` class is used to read and parse UBX, NMEA, and RTCM3 messages from different data streams.

```
!pip install pyubx2
from pyubx2 import UBXReader
```

4.1 Power and General Observations

The first analysis performed on the UBX file was checking when the receiver's time actually started to get spoofed. For this, we can examine the GNRMC message in the .ubx file (GN stands for GNSS and RMC stands for Required Minimum Specific) (Tavotech, n.d.). This message is part of the NMEA 0183 protocol which is used for communication between marine electronics and GNSS receivers. It provides essential GNSS data such as time, date, position, speed, and course overground. This information is crucial for navigation and tracking purposes. A sample GNRMC message is shown from the ubx file, giving all the crucial information:

```
['071745.00', 'A', '6916.54516', 'N', '01558.07361', 'E',
'0.013', '', '120924', '', '', 'A', 'V'], '_checksum': '1B',
'time': datetime.time(7, 17, 45), 'status': 'A', 'lat':
69.2757526667, 'NS': 'N', 'lon': 15.9678935, 'EW': 'E', 'spd':
```

```
0.013, 'cog': '', 'date': datetime.date(2024, 9, 12), 'mv':
'', 'mvEW': '', 'posMode': 'A', 'navStatus': 'V'}
```

The yellow highlighted part in the message shows the timing information, the green one shows the location, and the blue one shows the speed. If the timing information showed any jumps in the GNRMC message, we could deduce that the receiver's time was spoofed. Since our data starts at 7:17:45 UTC when the spoofing power was 0 dBm, I assumed there was no spoofing at that time. A true time series was created and the difference between that and the GNRMC message timestamps were observed as shown in the graph of Figure 14:

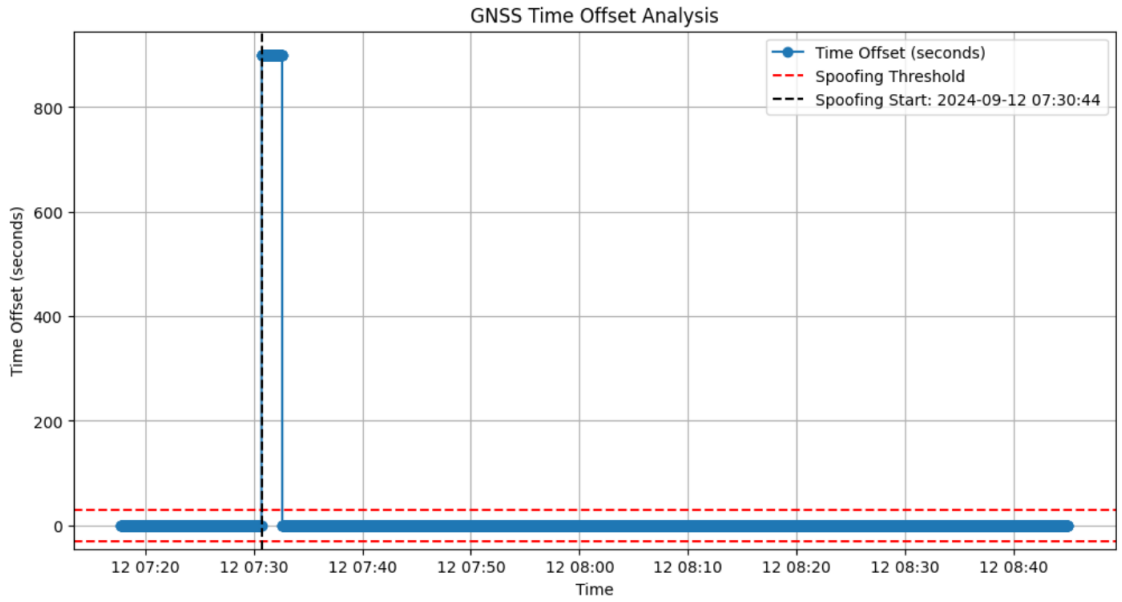

Figure 14: Time jumps in the GNRMC messages

There are a total of 5235 GNRMC messages in the ubx file. The spoofing threshold is set to be 30 seconds since the test log clarifies a time jump of 15 minutes (900 seconds). As we can see from the graph in Figure 14, the receiver started to get spoofed at around 2024-09-12 07:30:44, ending at 2024-09-12 07:32:37 for a total duration of 1.88 minutes. We can further zoom in on the graph and make a power analysis to see how much spoofing power was required to get the receiver time spoofed.

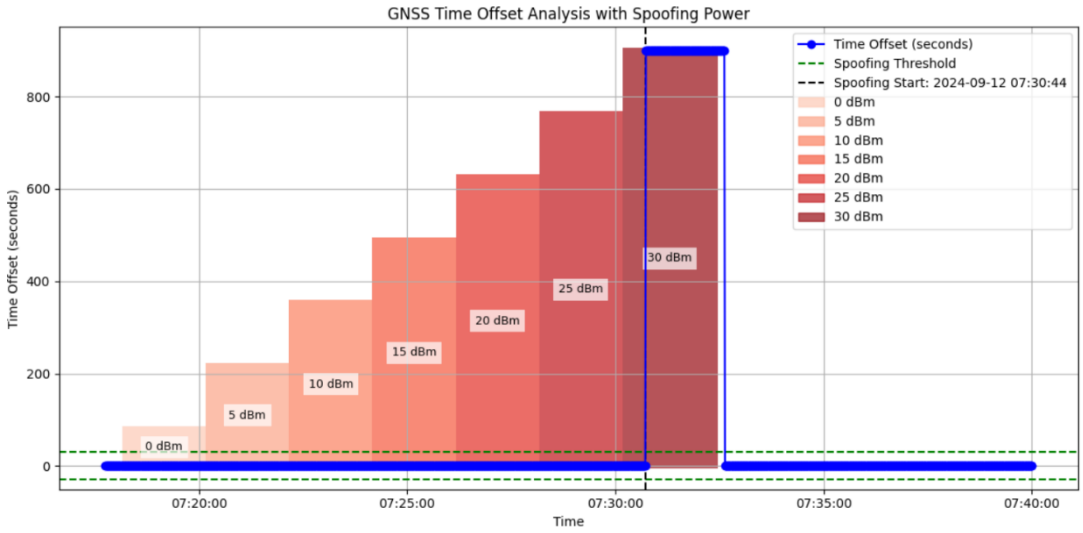

Figure 15: Graph showing how different spoofing power affected the receiver, clipped until 7:40 UTC.

In Figure 16 we can see that the u-blox f9p receiver showed excellent resilience against the time spoofing signal, only getting spoofed when the spoofing signal was at maximum power of 30 dBm. From the test log, we can see that 30 dBm power spoofed signal started at around 7:30:10 UTC and the receiver took approximately 34 seconds to get spoofed at that power. When the spoofer stopped working at 7:32:27, it took the receiver 10 seconds to return the time to normal state at around 7:32:37.

Next, if we want to see how the position and speed were affected, we can again take a look at GNRMC messages. Latitudes and longitudes can be converted into north and east coordinates using a local tangent plane approximation. The north component represents the distance in the north-south direction, while similarly, the east component represents the distance in the east-west direction, adjusted by the cosine of the reference latitude to account for meridian convergence. This way, we can visualize them in a better way as conversion to north and east allow us to check for subtle position shifts in meters and see how the location was affected in what was otherwise a static event. The formulas for conversion are given in equation 4 and 5:

$$\text{North} = \left(lat - lat_{ref}\right) \cdot \frac{\pi}{180} \cdot R_{earth} \tag{4}$$

$$\text{East} = \left(lon - lon_{ref}\right) \cdot \frac{\pi}{180} \cdot R_{earth} \cdot \cos\left(lat_{ref} \cdot \frac{\pi}{180}\right) \quad (5)$$

Where,

- lat and lon are the latitude and longitude of a point
- lat_{ref} and lon_{ref} are the reference (starting) latitude and longitude
- R_{earth} is the Earth's radius (6,378,137 meters)
- $\frac{\pi}{180}$ converts degrees to radians

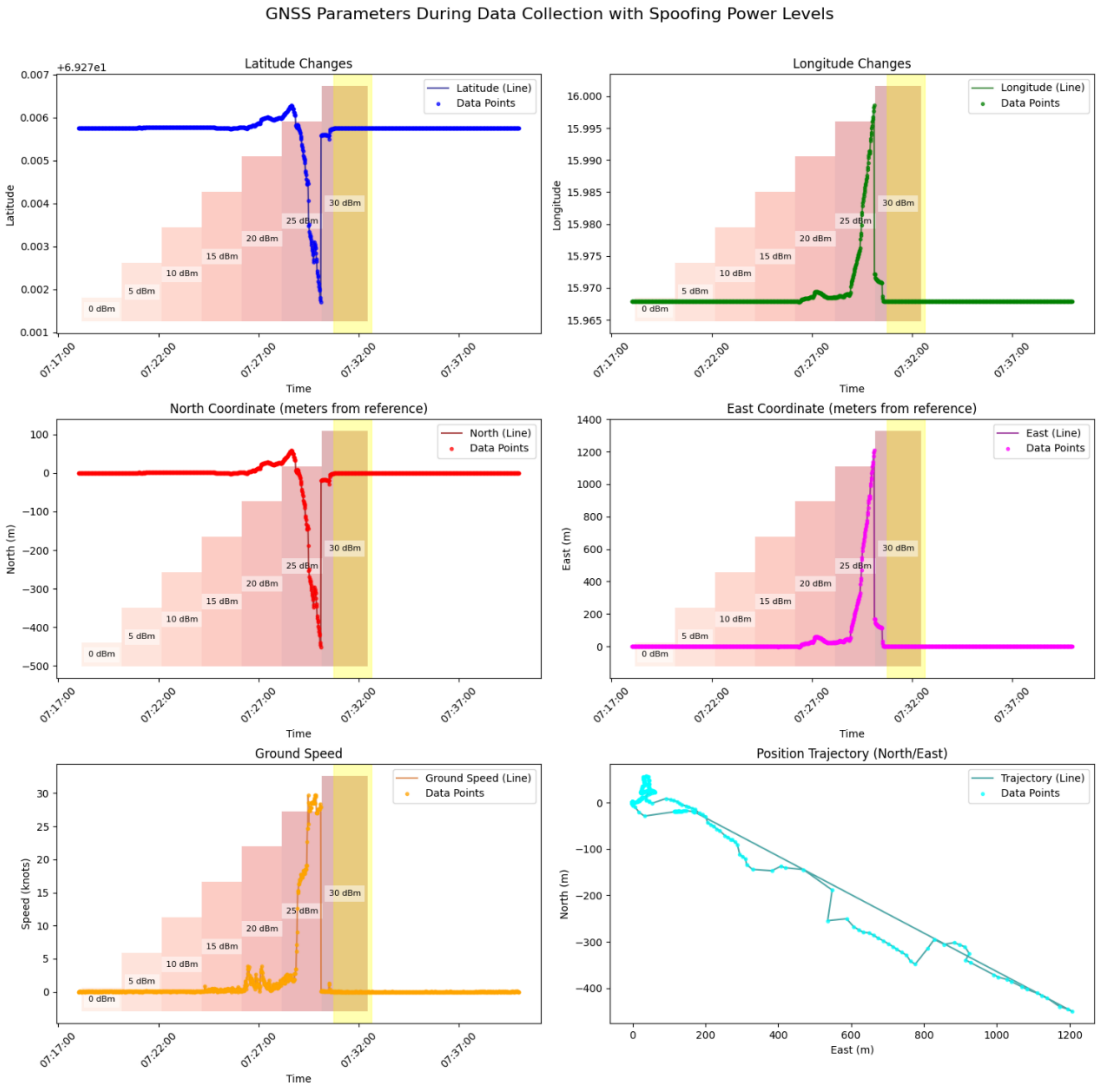

Figure 16: Location parameter change during the spoofing event. Yellow indicates when receiver experienced time jump.

The graphs in Figure 16 show the changes in various GNSS parameters during the 2.4.2 event. The yellow-shaded area indicates the period when the receiver's time was actually spoofed. An interesting observation is that even before the timing was spoofed, the receiver's location and speed had already been spoofed and showed changes. Once the receiver completely lost authentic signals, it started to broadcast last known fix hence why the location and speed looks recovered. The receiver suddenly appeared to travel approximately 1200 meters east and 400 meters south within one minute, reaching a maximum speed of 30 knots with a sudden stop, something that is not physically possible for a human or even a car unless it is an accident. These positional jumps happened when the spoofer was transmitting signals at around 20 – 25 dBm. The cause of this behaviour can be better understood by analysing the pseudorange graphs (see next section for more details). Another interesting observation is that the receiver has data points (taken from GNRMC) during the spoofing window, indicating the location, however same could not be said about the RAW messages that were used to derive this position. This discrepancy is also further explained in the next section.

Next, we can take a look at the HDOP (horizontal dilution of precision). HDOP is a measure of how satellite geometry affects the accuracy of GNSS horizontal positioning (Kaplan & Hegarty, 2006, p. 327). Lower HDOP values indicate better satellite geometry and more accurate positioning. This means more satellites are in Line of Sight (LOS) of the receiver and it is receiving better signal quality. As Kaplan and Hegarty (2006, p. 328) explain, HDOP represents the ratio between horizontal position error and the User Equivalent Range Error (which is standard deviation of pseudorange measurement errors (Kaplan & Hegarty, 2006, p. 327)), providing a numerical indicator of positioning reliability. The HDOP is extracted from GNGGA message which is also another NMEA message format containing information about HDOP and number of satellites used in location estimate (NovAtel, n.d.-a)

Coming to our graph, we can clearly see a rise in HDOP values when the receiver is getting spoofed. This is happening because the receiver is not catching signals from

satellites but instead from the spoofer, making the value of HDOP as high as 99, which practically means that no satellites were being read during that time. Even before the receiver time is spoofed at 7:30:10 UTC, the location and speed are already getting spoofed as shown in Figure 16, something which can verified from the HDOP graph in Figure 17. Once the spoofing period ends, we observe HDOP values returning to normal, indicating that now receiver is getting normal signals.

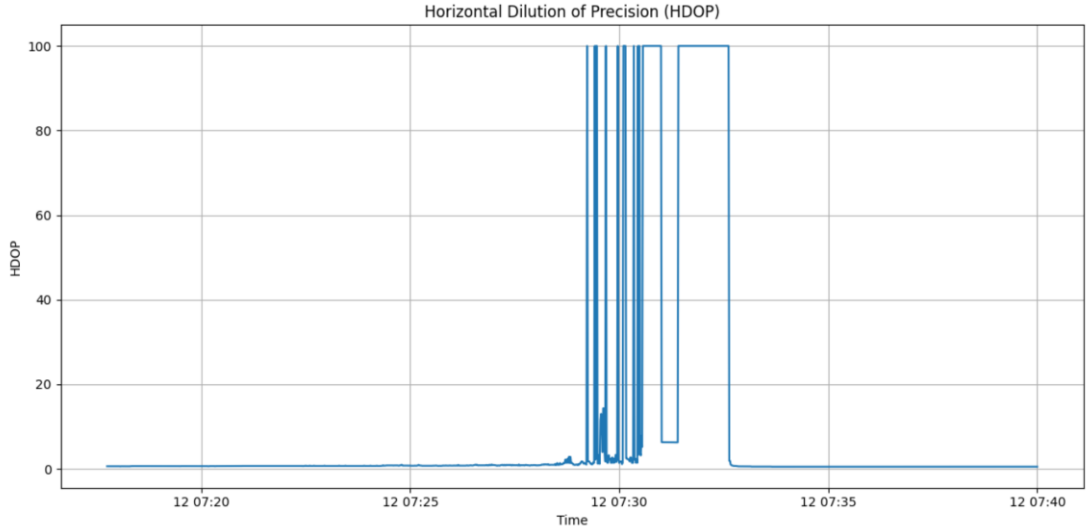

Figure 17: HDOP graph with respect to time

These are all the behaviours observed on the navigation layer of the message. In the next section, the raw data its behaviour is studied during the timing spoofing event.

4.2 Pseudoranges and Other Raw Observations

Pseudorange is the measure of physical distance between a satellite and a receiver which is used to estimate the location of the receiver. It is called "pseudo" because it includes various errors such as satellite clock errors, atmospheric delays, and multipath effects (He et al., 2020) (Yuanfa, Xigang, & Huli, 2009). It is one of the most fundamental raw measurements that is used in position estimation. Other raw data include C/N0 (Carrier-to-Noise-Density ratio), carrier phase and the doppler shift.

C/N0 is a measure of signal strength that indicates the quality of the received GNSS signal, expressed in dB-Hz. It represents the ratio of carrier power to noise power spectral density, higher the value, the better the signal quality (Ma et al., 2024). Carrier phase is the measurement of the phase of the incoming satellite signal's carrier wave, providing significantly more precise measurements than pseudorange but by resolving the integer ambiguity of the carrier cycles (Feng et al., 2012). Doppler shift is the frequency change in the received satellite signal due to the relative motion between the satellite and receiver, which can be used to determine velocity and assist in position calculations (Rouan, 2023).

These four measurements work together to provide comprehensive positioning solutions. Navigation algorithms typically integrate all these measurements through techniques like Kalman filtering to produce optimal position, velocity, and timing solutions while minimizing the impact of measurement errors and environmental interferences (Tondaś et al., 2023).

Now coming to our event 2.4.2, we can look at all the raw measurements one by one during the timing spoofing attack. For this we will check the RXM-RAWX message which contains the raw data. In the raw message format, pseudorange is denoted by C1C, C/N0 is denoted by S1C, carrier phase is denoted by L1C, and finally Doppler by D1C. In order to extract data and make streamlined plots, a Receiver Independent Exchange Format (RINEX) file was created from the .ubx file containing these four measurements. RINEX format serves as the standardized data structure utilized worldwide for sharing and processing GPS observations collected from the extensive network of IGS tracking stations, enabling diverse international research efforts in geodesy and atmospheric science (Jin, 2012, p. 360). Since many data points were either missing or had unrealistic values, libraries like `georinex` or software such as RTKLIB couldn’t be used, and instead custom Python functions were written to parse the RINEX file.

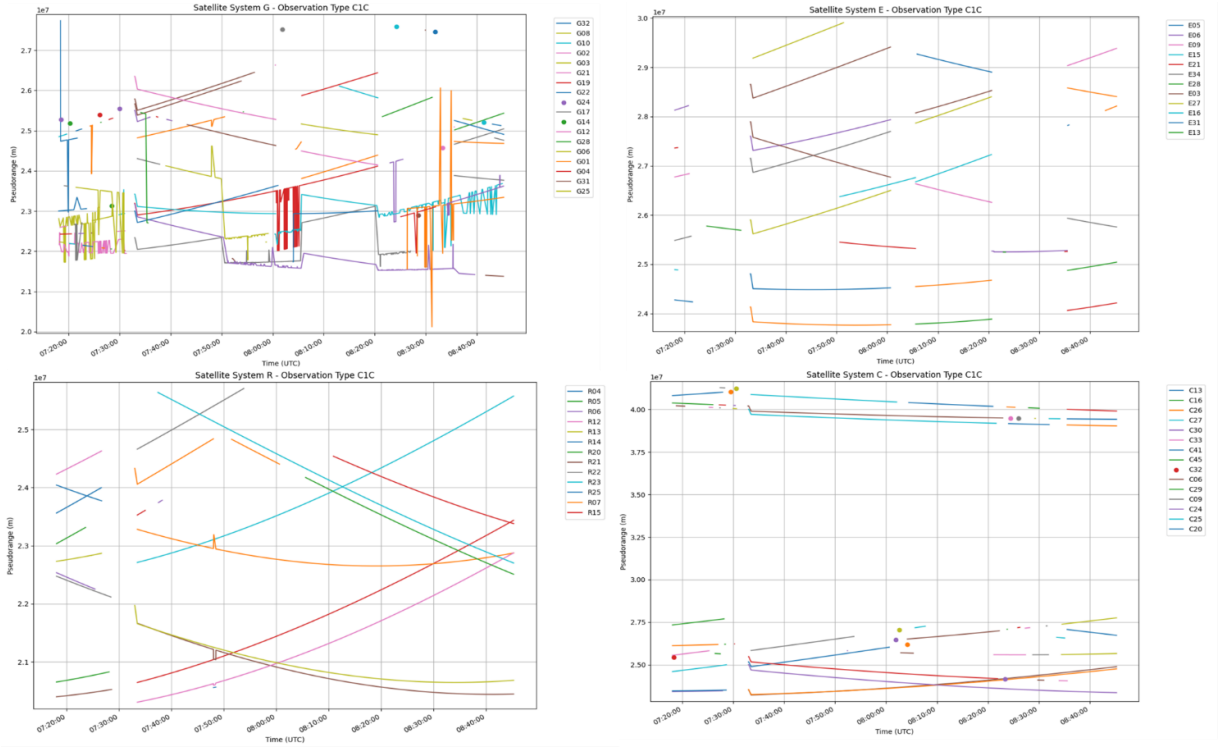

Figure 18: Pseudoranges of the whole ubx file showing all satellite systems. From left to right in first row it shows GPS and then Galileo. In the second row it shows GLONASS and BeiDou.

Figure 18 depicts the C1C observations or pseudoranges for all four satellite systems during the entire duration of the data collected in the ubx file. G is for GPS, E is for Galileo, R is for GLONASS, and finally C is for BeiDou. QZSS is ignored during the analysis due to its unavailability during the spoofing event in the geographical location at Bleik. Since the spoofer was mostly affecting the GPS and Galileo satellite systems, we can see that these are mostly affected. According to the jammertest log (2024-a), event 2.4.2 ended at around 7:32 UTC, so we would be mostly focusing on the disturbances caused during that time. In the whole diagram, we can see the effects of jamming and pseudorange errors also occurring later in the ubx file, but they are out of scope of this thesis.

Now firstly let's zoom in on the C1C observations that were recorded during the 2.4.2 event as shown in the Figure 19.

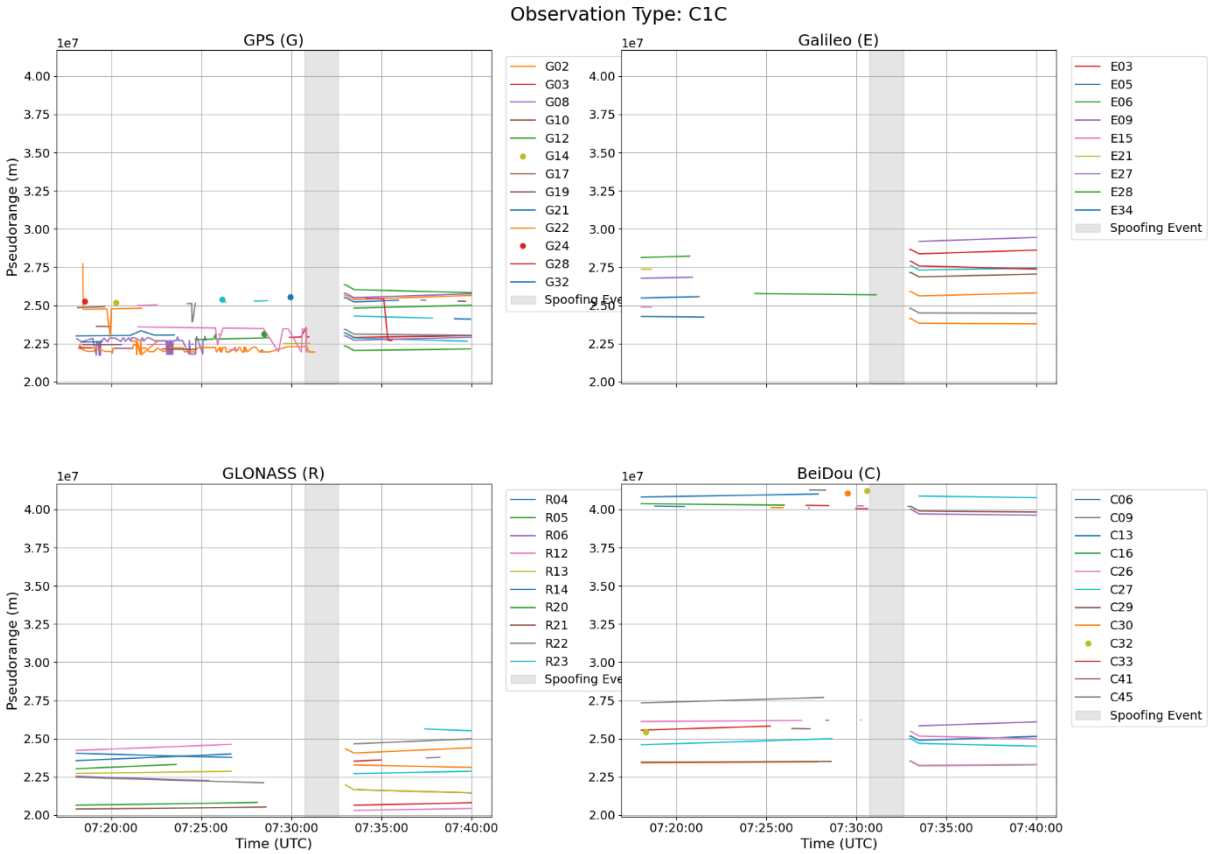

Figure 19: Pseudoranges during the 2.4.2 spoofing event

The first observation we can make from Figure 19 is obvious: the spoofer was actively preventing the receiver from connecting to the satellites, leaving it void of any useful raw data during the spoofing event. As seen in the shaded area, which represents the time when the receiver's time was spoofed, only a few measurements from GPS and Galileo were recorded. The GLONASS system stopped working even a few minutes earlier than the time jump, approximately when the location was being spoofed. As for BeiDou, a few satellites such as C32 were visible close to the time jump, but as soon as the time jump occurred, they also disappeared. When the spoofing was turned off, the receiver recalibrated itself, and within 10 seconds all the satellite constellations were visible again, allowing the receiver to obtain a location fix.

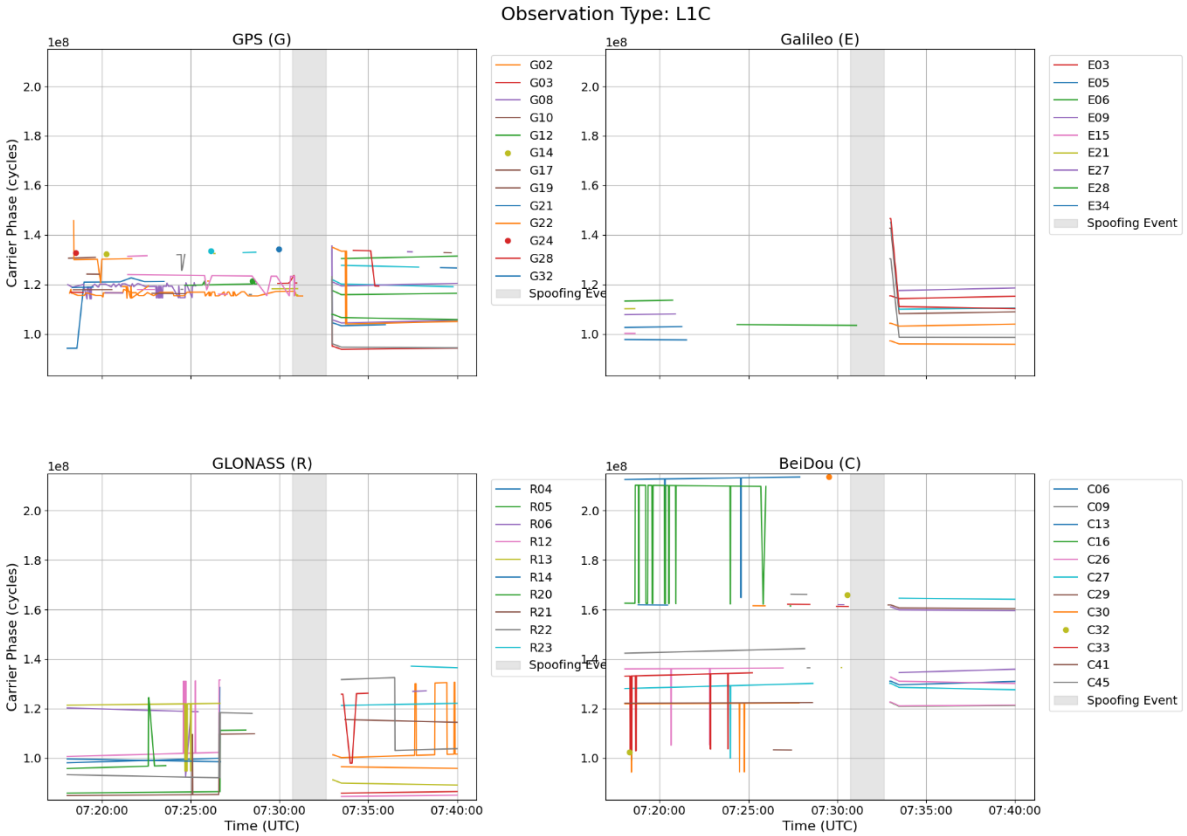

Figure 20: Carrier Phase during the 2.4.2 spoofing event

Just like the pseudorange graphs, carrier phase shows a similar trend. For GPS and Galileo, they have multiple stable carrier phase measurements before the spoofing event, and as the raw data becomes unavailable during the time jump, so do the measurements of carrier phase. For GLONASS and BeiDou, both show greater carrier instability and erratic behaviour before the time jump, especially BeiDou showing cycle slips in C16 and C33. These cycle slips occur when a receiver temporarily loses lock on the satellite signal's carrier wave and then reacquires it. When this happens, the receiver's phase tracking loop can't maintain count of the exact number of complete wavelengths between the satellite and receiver (Hu & Fang, 2009). This is likely caused by fluctuating signal strength (something could be blocking it) as seen earlier in the C/N0 graphs, or the satellite geometry since these satellites appear to be furthest as indicated by the pseudorange graphs. Once the spoofer is turned off, everything returns to normal except a few fluctuations in carrier phase in GLONASS which could be due to dropping signal strength.

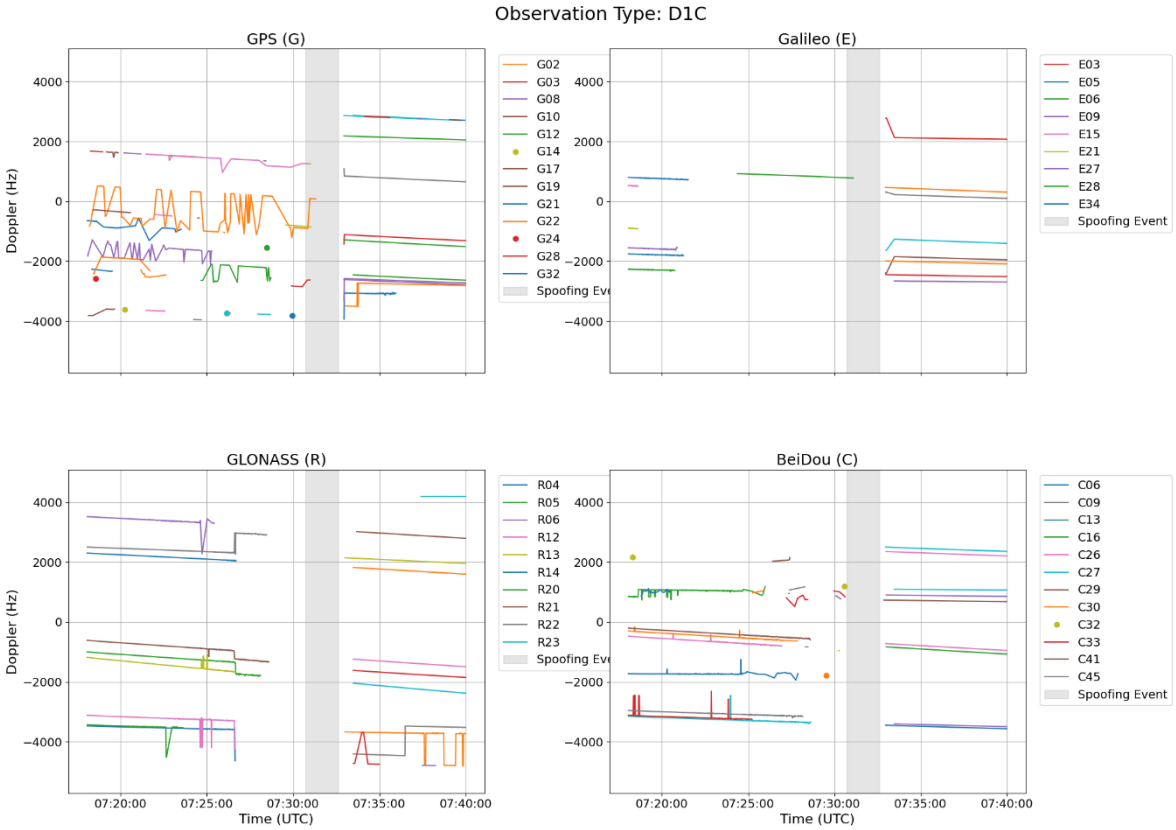

Figure 21: Doppler shift during the 2.4.2 spoofing event

Lastly, observing the Doppler shift graphs, we again see a similar trend. Data is mostly unavailable during the time jump. GPS shows some oscillating patterns, indicating changing relative velocities of the satellites. Galileo has fewer visible satellites but shows more stable Doppler measurements than GPS. For GLONASS, some signals show abrupt changes before disappearing, a likely indication that something is wrong. For BeiDou, some satellites exhibit unusual patterns with rapid changes and frequent jumps, again due to low signal quality and cycle slips as seen in the carrier phase graphs. Once the spoofer is turned off, the Doppler shift returns to normal except for a few fluctuations in GLONASS in a satellite that is receding from the receiver.

As indicated in the section 4.1, the receiver does have location despite not having the raw data available during the time jump. When observing the location and speed, the

receiver is simply outputting the last known value, hence, creating an illusion of position despite not having any data. In the next chapter we can see that the receiver was marking the solution as invalid (a traditional way of detecting spoofing or simply solution availability limitation due to degraded signal environments such as indoor or tunnels etc.), so whatever position solution it was outputting was simply of no use. Although the plots show complete lack of data during the known time jump period, some satellites (either 1 or 2 at times) were still visible. They were just simply not enough to compute any position since a minimum of four satellites are needed to compute location and time.

5 Traditional Detection

As explained in the chapter 2, there are many ways to detect if the receiver is spoofed. The method that can be relied on with the u-blox receiver is its validity flag illustrating solution availability and trustworthiness. In this chapter we look at the validity flag and pseudorange RMS errors and the C/N0 to check if the receiver was correctly identifying the interference. Referring to the flowchart in Figure 5, the detection would be studied using signal analysis method (C/N0), validity flag and the positioning method (pseudorange RMS errors).

5.1 Signal Analysis Method

The signal analysis method used is a power analysis approach or more precisely a method focusing on looking at the C/N0 values of the incoming signals.

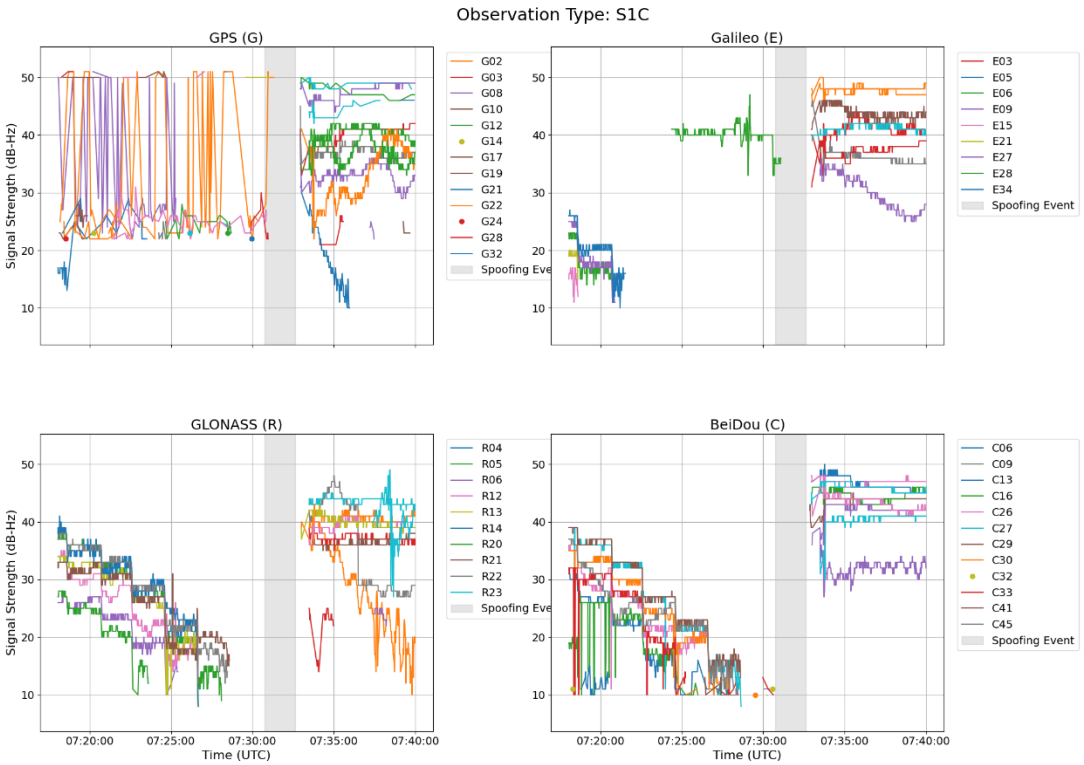

Figure 22: C/N0 during the spoofing period

Looking at the C/N0 graph in Figure 23, a big gap can be observed during the spoofing period. The GPS signal strength remained within 20-50 dB-Hz for event 2.4.2. Satellite G02 showed the highest fluctuation, ranging from 22 to 52, even achieving a high signal strength of 50 during the time jump but quickly fading afterwards when the power of the spoofer was too high. Similarly, E05 from Galileo was the only satellite showing some resilience during the time jumps, but its signal also died down once the receiver started to experience the time jump. For the Russian GLONASS and Chinese BeiDou, both showed similar patterns, with their C/N0 declining as the spoofer's power increased, eventually disappearing altogether a few minutes before the time jump. They also had relatively low signal strength compared to the other two satellite systems. Once the spoofer was turned off, just like before, every satellite system returned to normal within 10 seconds. Next, we can check the carrier phase measurements.

As stated by Radoš et al. (2024), the C/N0 tends to increase during the spoofing period. Figure 24 shows the sparse values of the satellites raw data during the spoofing period.

2024-09-12 7:30:51	G	2	22308982.07	117234491.9	-901.484	30
2024-09-12 7:30:51	G	17	23450433.96	123232862	1257.827	26
2024-09-12 7:30:51	G	21	22937332.43	120536494.5	-2630.479	23
2024-09-12 7:30:52	E	28	25693500.53	103457079.2	781.193	35
2024-09-12 7:30:52	G	19	22506903.93	118274578.2	-850.807	50
2024-09-12 7:30:52	G	21	22937828.59	120539102	-2630.479	22
2024-09-12 7:30:53	E	28	25693306.5	103456298	781.197	35
2024-09-12 7:30:53	G	3	22021653.82	115724571.2	1255.725	50
2024-09-12 7:30:53	G	19	22507066.51	118275429.3	-852.178	50
2024-09-12 7:30:54	E	28	25693112.67	103455517.2	780.766	35
2024-09-12 7:30:54	G	3	22021415.82	115723315.3	1256.006	50
2024-09-12 7:30:54	G	19	22507228.84	118276281.2	-852.036	50
2024-09-12 7:30:55	E	28	25692918.81	103454736.7	780.317	35
2024-09-12 7:30:55	G	3	22021171.35	115722059.7	1255.377	50
2024-09-12 7:30:55	G	19	22507391.01	118277133.7	-852.727	50
2024-09-12 7:30:56	E	28	25692725.02	103453956.4	779.972	35
2024-09-12 7:30:56	G	3	22020936.15	115720804.6	1254.742	50
2024-09-12 7:30:56	G	19	22507553.39	118277986.8	-853.628	50
2024-09-12 7:30:56	G	21	22939837.13	120549656.9	-2630.479	22

Figure 23: Raw measurements sample with C/N0 highlighted. These are in dB/Hz.

Judging from the given values in Figure 24, mostly the GPS satellites consistently maintain a high signal to noise ratio values. While this could be a potential indication of interference, the lower value from other satellites throws of the entire reading and makes it difficult to conclude whether that is a result of interference or just naturally good signal strength at the time. Therefore, it can be safely concluded that signal to noise ratio is not very helpful in detecting time spoofing attempt in this case.

5.2 Positioning Method

Next, we can look at the pseudorange RMS errors which is a part of a positioning-level related detection method. The RMS error in pseudorange measurements represents the statistical deviation between expected and actual signal readings. These discrepancies stem from multiple sources including satellite clock inaccuracies, ionospheric and tropospheric errors, and equipment-related noise factors. Spoofing detection methodologies can identify potential threats by examining variations in pseudorange measurements across consecutive time intervals, revealing anomalous patterns that would indicate signal manipulation attempts (Shang et al., 2022) (Angrisano et al., 2013). RMS errors of the whole .ubx file is plotted in Figure 24.

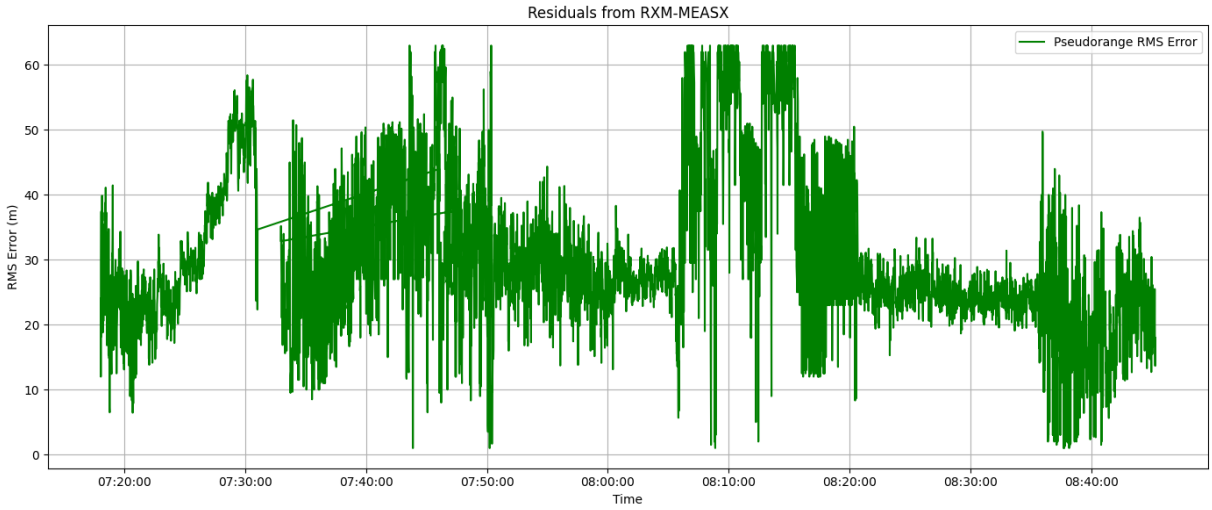

Figure 24: Averaged pseudorange RMS errors in the ubx file

These errors were extracted from RXM-MEASX messages, which is another format of raw data available within a u-blox receiver. Upon closer examination of the graph in Figure 25, we notice straight lines at our time jump points. Since we have already analysed the raw measurements, it can be concluded that this is due to the unavailability of raw data. These straight lines, although resulting from a lack of measurements, can also indicate that something is wrong, and the data gathered during this period is likely invalid.

5.3 Validity Flag Method

For even better understanding, next the validity flag is studied which is a part of NMEA analysis method. Figure 26 shows the data validity plot from 7:18 UTC to 7:40 (duration of the 2.4.2 event).

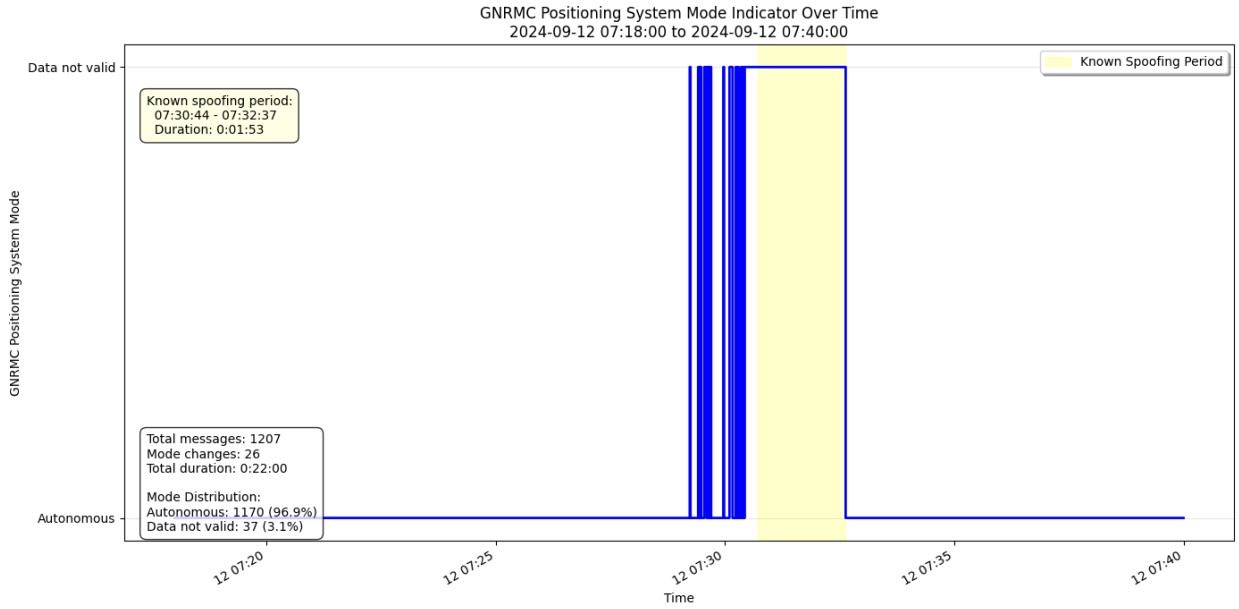

Figure 25: GNRMC message status

The validity flag was extracted from the GNRMC messages. In the GNRMC message, "A" or "Autonomous" stands for a valid solution and "V" is for an invalid solution. In Figure 26, we can see that the u-blox receiver was correctly identifying the spoofing region. It marked the whole region as invalid and was 100% detecting that something was wrong. However, looking at the number of parsed messages, something peculiar is noticed. The receiver is missing data for almost about two minutes, the same time when the time jump was valid. Since Figure 25 is a continuous plot, those missing points are also marked as invalid. If the receiver had moved indoors and GNSS data became unavailable, it would also have been marked invalid. Another thing we notice is that the receiver started to mark messages as invalid even before the time jump occurred. This is due to the fact that location and speed were affected before the time was spoofed. However, it is still giving some valid messages, struggling between validity and invalidity. If this is taken as an indicator for time spoofing in many applications where timing is only required and location/speed aren't needed (such as in grid time synchronization), these fluctuations before the time jump could act as an early warning, even though they are false positives if only taking timing spoofing into consideration. In the case of event 2.4.2, location

spoofing had no effect on receiver’s time. Once the spoofing period is over, the validity flag returns to normal after receiver gets a position fix again.

Even though the receiver is successful in detecting the spoofing region, it can still be enhanced using AI to detect the timing spoofing and mark even those points as invalid that it didn't have enough data to compute location from. As mentioned in the chapter 3 and shown in Figure 24, some satellites were still visible during the time jump even when the receiver wasn't outputting any validity flag or location, just repeating the last known fix. The chapter 6 will talk about detection using AI and how it can further enhance the detection.

6 AI based Detection

In this section, the AI based detection system is introduced, and results are explained.

6.1 Isolation Forest

Isolation forest is chosen for AI based detection. The Isolation Forest algorithm, first developed by Liu, Ting, & Zhou (2008) introduces an innovative approach to anomaly detection that fundamentally differs from conventional methods. Isolation Forest operates as an unsupervised learning algorithm, which is crucial for GNSS security applications. Rather than creating profiles of normal behaviour, it specifically targets and isolates anomalies directly. This method capitalizes on the inherent nature of anomalies as they appear infrequently and possess distinctive characteristics compared to normal data points (Liu, Ting, & Zhou, 2008). The technique constructs random decision trees that recursively partition data using randomly selected attributes and split values. Since anomalies typically have unusual feature values and occur rarely, they tend to be isolated much closer to the root of these trees, requiring fewer partitioning steps. By building an ensemble of these trees, the algorithm identifies potential anomalies as those instances consistently showing shorter path lengths across multiple trees. This approach enables Isolation Forest to achieve remarkable efficiency with linear time complexity and minimal memory requirements, making it particularly valuable for analysing massive datasets and handling high-dimensional data where traditional distance calculations become prohibitively expensive (Liu, Ting, & Zhou, 2008).

Isolation forest does not require labelled examples of every possible spoofing attack, making it effective at detecting novel spoofing techniques that weren't present in training data. This is particularly important in the cybersecurity domain where attackers constantly evolve their methods. According to Mohammed et al. (2024), when compared against other algorithms for detecting GPS spoofing in UAVs, isolation forest had the highest accuracy, 95.85% as compared to LightGBM (95.23%), Random Forest (93,63%),

XGBoost (95,52%) etc. Isolation Forest also requires relatively few hyperparameters to tune. The main parameters (number of trees, subsampling size, and contamination rate) have intuitive interpretations and can be optimized efficiently. Another reason for choosing an isolation forest is because it can be easily scaled through parallel processing since each tree in the forest is independent. This makes it adaptable to various hardware platforms, from resource-constrained embedded systems to more powerful computing environments. Another reason for choosing isolation forest is that is still very rarely explored in terms of time synchronization attack scenarios like this. Using it on jammertest data will provide insight if it is a good choice in combating GNSS timing spoofing.

The goal of having an AI based detection system is not to replace the internal detector but to enhance the detection and make sure even the instances that were missed by receiver can be correctly identified as interference.

6.2 Training, Testing and Validation

In order to implement the isolation forest algorithm, a CSV file was created from the RINEX and the .ubx file having 15 classes as shown here.

```
timestamp,     constellation, satellite,     C1C, L1C, D1C,
S1C, spoofing, time_jump,     satellite_id,  HDOP,     lati-
tude,     longitude,validity, speed.
```

The spoofing and time jumps were just indicators that the spoofing was turned on and a positive value of time jump indicated that the receiver's time was actively spoofed during these specific messages. Once the CSV was ready, the data was split into training, testing and validation sets, with training on 60% of the data and using 20% for testing and validation. The number of samples in each set are shown here.

```
Train set: 14124 samples, Anomalies: 43 (0.30%)
Validation set: 4709 samples, Anomalies: 14 (0.30%)
Test set: 4709 samples, Anomalies: 14 (0.30%)
Clean train set: 14062 samples (removed 62 outliers)
```

The outliers in the training set were removed using the Z-score method (Aggarwal et al., 2019). In the training set, data points were kept unless there were more than 4 standard deviations from the mean. With normally distributed data, this would retain approximately 99.99% of the data. The reason for such a conservative approach was due to lack of spoofed data (only 43 anomalies out of 14124 data points) and I wanted to retain as much data as possible.

Now that our data was split into appropriate sets, we can build an algorithm for isolation forest implementation. First comes the creation of trees. For a dataset $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$ where each x_i represents a GNSS signal with *d* features (C1C, L1C, D1C, S1C, etc.), the algorithm creates a forest of random isolated trees.
For each tree:

- A random subsample $X' \subset X$ of size $\psi \cdot |X|$ is selected, where $\psi \in (0,1]$ is the *max_samples* hyperparameter.
- The tree recursively partitions the space by:
  i) randomly selecting a feature $j \in \{1,2, \dots, d\}$ from a random subset of $\lambda \cdot d$ features where, $\lambda \in (0,1]$ is the *max_features* hyperparameter
  ii) having a random split value p with the minimum and maximum values of feature j
  iii) creating a partition: $X_{\text{left}} = \{x \in X' | x_j < and X_{\text{right}} = \{x \in X' | x_j \geq p\}$
  iv) continuing recursively until either a single point is isolated, or a maximum height is reached.

Once the trees are made, we move onto the calculation of the path length. The path length $h(x)$ for a point x is defined as the number of edges traversed from the root node to the terminating node in an isolation tree. For a forest with t trees (determined by the *n_estimators* hyperparameter), the average path length is given in equation 6:

$$H(x) = \frac{1}{t}\sum_{i=1}^{t} h_i(x) \tag{6}$$

where $h_i(x)$ is the path length of x in the i-th tree (Liu, Ting, & Zhou, 2008).

Then we proceed to the anomaly score formulation. We used time jump as an anomaly. For a dataset of size n, the expected path length of unsuccessful search in a Binary Search Tree is:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{7}$$

where $H(i)$ is the harmonic number $\sum_{j=1}^{i} \frac{1}{j} \approx \ln(i) + 0.57721$ (Euler's constant).

The anomaly score *s* for an instance x is defined in equation 8:

$$s(x,n) = 2^{-\frac{H(x)}{c(n)}} \tag{8}$$

where:

s is close to 1 for anomalies (shorter paths)

s is close to 0 for normal instances (longer paths)

In my implementation, I used the negative of this score as the anomaly score, so higher values indicate more anomalous points.

Next, the contamination parameter is needed. The contamination hyperparameter $\alpha \in (0,0.5)$ represents the expected proportion of anomalies in the dataset. This parameter influences the threshold selection during model training:

$$\theta_{\text{initial}} = \text{Quantile}(\{-s(x_i,n) | x_i \in X\}, 1-\alpha) \tag{9}$$

Now that the contamination parameter has been obtained, the decision threshold parameter also needs to be determined. The decision boundary for classifying a point as an anomaly (time spoofing) is determined by a threshold θ on the anomaly score:

$$f(x) = \begin{cases} 1 & \text{if } -s(x,n) > \theta \text{ (anomoly)} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The optimal threshold $\theta^*$is determined by maximizing a scoring function on a validation set *V* as shown in equation 11:

$$\theta^* = \text{argmax}_\theta \text{Score}\big(y_V, f_\theta(X_V)\big) \tag{11}$$

where Score is the F1-score as given in equation 12:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

with:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

This is the base of our isolated forest algorithm (Google Developer, 2025). Now in order to improve the accuracy, we need to perform hyperparameter tuning. For that we need to define the bootstrap parameter and feature importance.

The *bootstrap* hyperparameter $\beta \in \{True, False\}$ determines whether the subsamples for each tree are drawn with replacement:

$$X_i' = \begin{cases} SampleWithReplacement(X, \psi. |X|) & \text{if } \beta = True \\ SampleWithoutReplacement(X, \psi. |X|) & \text{if } \beta = False \end{cases} \tag{14}$$

For each feature *j*, importance is calculated based on how frequently it's used for splitting across all trees:

$$I(j) = \frac{1}{t}\sum_{i=1}^{t}\sum_{k\in S_i} 1(f_k = j)\cdot \Delta\text{impurity}_k \quad (15)$$

where:

S_i is the set of split nodes in tree *i*

f_k is the feature used at split node *k*

$\Delta\text{impurity}_k$ is the change in impurity (often approximated by the number of points separated at node k)

The optimal hyperparameters $\Theta^* = \{t^*, \psi^*, \lambda^*, \alpha^*, \beta^*\}$ are determined through grid search:

$$\Theta^* = \text{argmax}_\Theta \text{F1}\big(y_V, f_\Theta(X_V)\big) \quad (16)$$

In this implementation, the best hyperparameters were calculated to be:

n_estimators (t^*): 100

max_samples (ψ^*): 0.7

max_features (λ^*): 0.9

contamination (α^*): 0.005

bootstrap (β^*): False

Now that the algorithm was ready, it was implemented on the training dataset with checking the results on testing and validation sets.

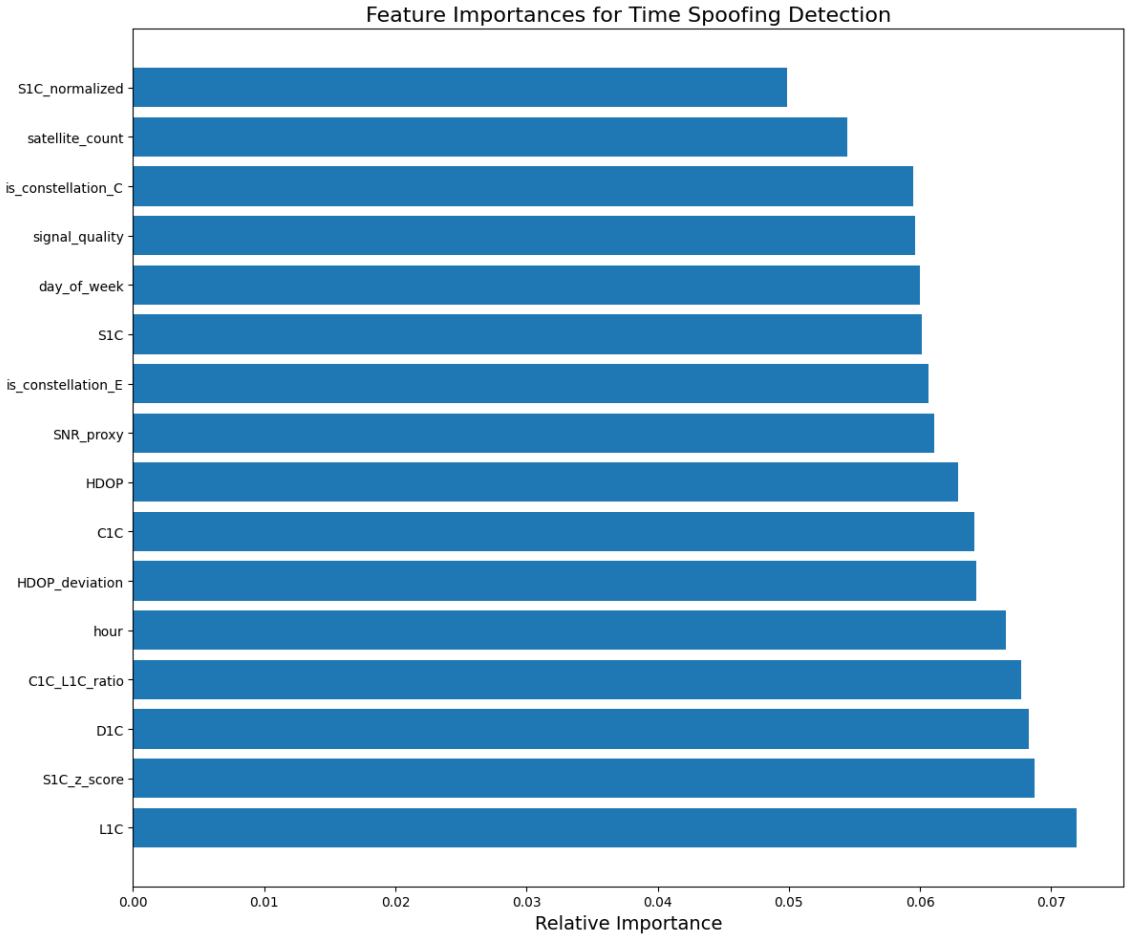

Figure 26: Feature importance for training dataset.

Figure 27 shows the features that were selected for training the dataset. These features were selected based on testing on the validation set after training since they produced the best results. As can be observed, the validity flag was not used in training Using the validity flag could cause data leakage since it's directly related to the internal detector's output, and I am trying to build an independent system that enhances rather than replicates it. Our model focuses on raw signal characteristics (L1C, D1C, C1C, S1C) and their derivatives because spoofing manifests first in these measurements. Moreover, during time jump events, location data is often completely missing. A model depending on location/speed would fail exactly when it is needed the most. Signal characteristics were preferred because they work regardless of whether the receiver is static or moving, while speed-based anomaly detection would need to know the expected motion profile. The section 6.3 will highlight the results achieved.

6.3 Results

The Isolation Forest algorithm worked better than expected given the scarcity of data, achieving an accuracy of 87%, detecting all the spoofing instances present in the test dataset. The Table 2 and 3 show the confusion matrix and results parameters.

Table 2: Confusion matrix of results

	Predicted 0	**Predicted 1**
Actual 0	4693	2
Actual 1	0	14

Table 3: Results parameters of implemented isolation forest

Metric	**Results**
Precision	0.8750
Recall	1.0000
F1-score (Class 1)	0.93
Specificity	0.9996
False Positive Rate	0.0004
Accuracy	1.00

Breaking down the confusion matrix,

- True Negatives (Actual 0, Predicted 0): 4693 cases were correctly identified as normal instances
- False Positives (Actual 0, Predicted 1): 2 cases of normal instances were incorrectly flagged as spoofing
- False Negatives (Actual 1, Predicted 0): 0 cases were missed in no spoofing instances
- True Positives (Actual 1, Predicted 1): 14 cases were correctly identified as spoofing instances

Now for the metrics:

- **Precision (0.8750)**: This shows that when the model flags something as spoofing (class 1), it's correct 87.5% of the time. This comes from $\frac{14}{14+2}$.
- **Recall (1.0000)**: The model caught 100% of all actual spoofing instances, it didn't miss any.
- **F1-score (0.93)**: This is the harmonic mean of precision and recall, indicating an excellent balance between the two metrics.
- **Specificity (0.9996)**: The model correctly identified 99.96% of normal instances, meaning very few legitimate cases were flagged as suspicious.
- **False Positive Rate (0.0004)**: Only about 0.04% of normal instances were incorrectly flagged as spoofing, which is remarkably low.
- **Accuracy (1.00)**: This seems to be rounded up from 0.9996, as the overall accuracy is $\frac{4693+14}{4693+2+0+14} = 0.9996.$

The most impressive aspect is the perfect recall (1.0) which means it was catching all spoofing attempts while maintaining very few false alarms. In security contexts like spoofing detection, this balance is ideal since it minimized both dangerous misses (false negatives) and annoying false alarms (false positives). These results also prove that choosing isolation forest for this task was a good choice. Once it is combined with the internal detector (validity flag), no spoofing instance will be missed. Figures 27 and 28 show various performance statistics of the AI model.

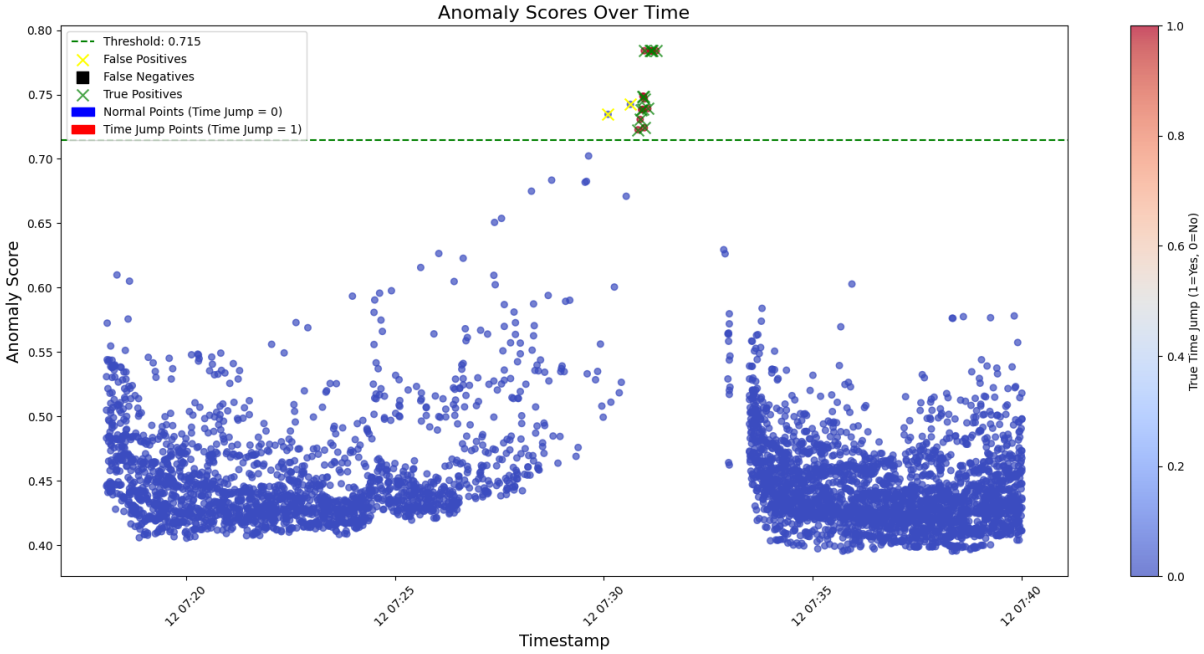

Figure 27: Anomaly score distribution

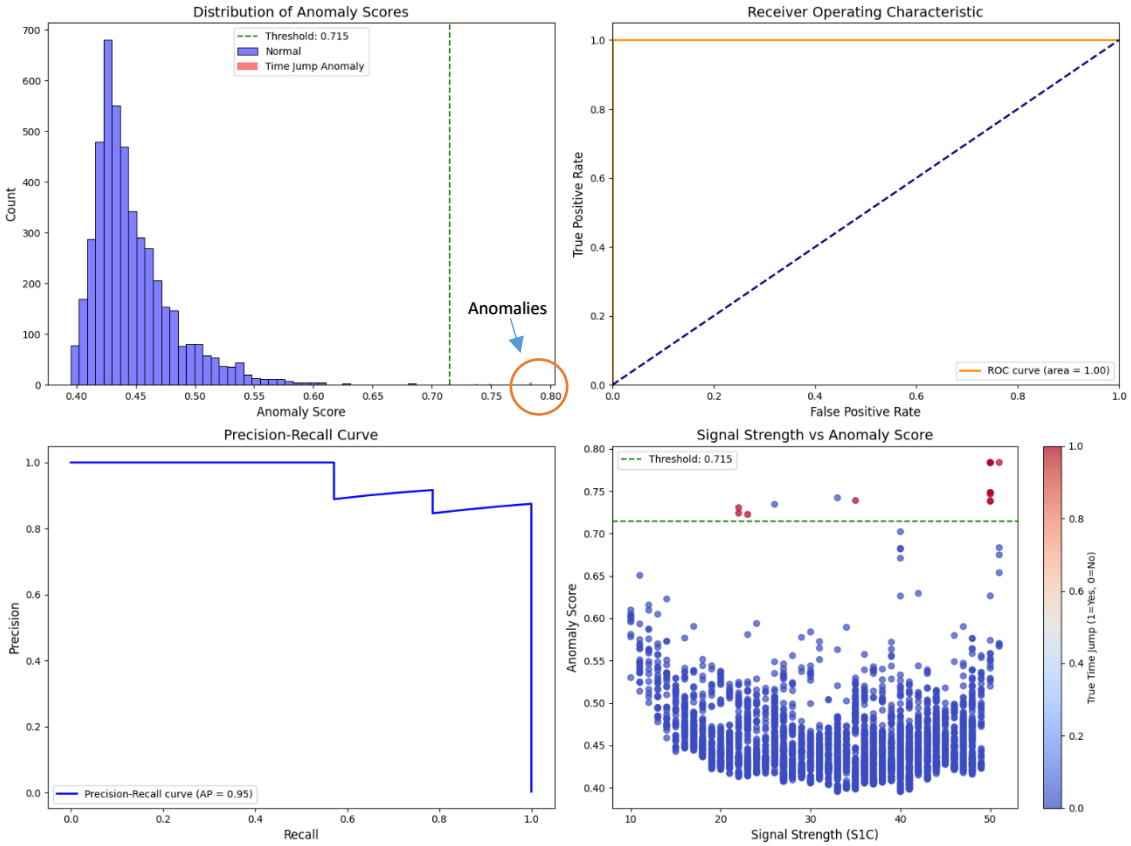

Figure 28: More result characteristics

The receiver operating characteristic (ROC) curve demonstrates the model's ability to differentiate between regular and irregular data patterns across various decision thresholds. Since the plotted curve sits substantially above the reference diagonal, it indicates that the model is performing much better than random chance at distinguishing between normal GNSS data and time jump anomalies. The Precision-Recall curve shows the trade-off between precision and recall. An average precision score of 0.95 means excellent performance at detecting time jumps accurately.

Due to lack of data, further testing of the isolation forest-based detection is not currently possible. Since the data is scares and limited, more experimentation is needed to confirm the ability of the isolation forest to catch time spoofing attempts. It is then combined with the internal detector and the performance metrices are shown Figure 29:

Figure 29: Enhanced detector based on combining AI and Validity flag

When applied the enhancement, the instances that were missed due to lack of data (having no validity flag and location but having one or two satellites still visible in raw data) were correctly identified. The zeros in Figure 29 is based on those missing instances. With the early warning system of validity flag and detection of spoofing on raw data

using AI, both result in a detection system that is sure not to miss any timing spoofing instance.

7 Implementation Considerations and Practical Applications

This chapter presents the discussion of the results, providing insight on how they can be implemented in real world situations.

7.1 Real World Integration Challenges

The integration of GNSS timing spoofing detection and mitigation systems into existing critical infrastructure presents several significant challenges that must be addressed for successful real-world implementation.

The first challenge faced is retrofitting. Integrating spoofing detection systems into existing infrastructure requires careful consideration of compatibility with legacy systems. Many critical infrastructure systems rely on older GNSS receivers that lack modern security features or the processing capability to implement advanced detection algorithms. This creates a technical gap where the most vulnerable systems, such as 5G networks with complex, higher-density synchronization needs, require more robust protection against GNSS jamming and spoofing (Inside GNSS, 2021). The challenge extends to designing retrofits that don't compromise the primary functionality of these systems. For new implementations, modern receivers must be designed with built-in security features, as highlighted by European GNSS Agency initiatives such as GIANO and GEARS projects which aim to make critical infrastructure, particularly energy networks, more robust against spoofing attacks (EUSPA, 2020). These new implementations must ensure that detection algorithms don't introduce unacceptable latency or consume excessive computational resources.

The next challenge is real time implementation. As observed in our Jammertest 2024 data analysis and supported by industry findings, timing spoofing attacks can manifest rapidly, with detection latency needing to be under three seconds for effective mitigation in critical applications (GPSPATRON, 2023). This real-time monitoring is essential for

critical infrastructure systems such as telecommunications networks, power grids, and financial transaction systems that require continuous timing synchronization with minimal interruption. The computational overhead of constant signal monitoring presents practical limitations, particularly for systems with constrained processing. While the implented isolation forest algorithm demonstrated high accuracy with relatively low computational demands, its integration requires careful optimization to ensure minimal impact on system performance.

Another one of the most significant challenges in deploying spoofing detection systems is managing false alarms. According to Inside GNSS (2020), applications of critical importance may justify expensive protective measures, as even improbable events can result in devastating consequences. This is why it must be addressed to prevent significant societal disruptions. As demonstrated in the results in Chapter 5 and 6, even high-performing algorithms such as the implemented isolation forest occasionally produce false positives (2 out of 4695 normal instances in our case) and the false alarms produced by the validity flag. Developing appropriate response protocols requires balancing detection sensitivity against operational continuity. This is particularly challenging in applications where timing accuracy is crucial, such as telecommunications networks where microsecond-level synchronization is required for proper functioning, as seen in 5G deployments that rely on precise timing from GNSS sources.

Implementing distributed detection systems across multiple receivers introduces synchronization challenges of its own. Monitoring and classification systems for GNSS interference require reliable communication channels and standardized reporting protocols for correlation of suspicious events across geographically dispersed receivers (GPSPATRON, 2023). While such systems offer improved detection capabilities through comparative analysis, they also introduce additional points of failure and complexity.

The trade-offs between centralized and distributed detection architectures must be carefully evaluated based on the specific requirements of each application domain.

Systems like TimePictra that combine BlueSky GNSS Firewall management with monitoring capabilities provide a unified view of the entire timing architecture and all timing sources, enabling operators to have full control over resilient timing architectures (Inside GNSS, 2021).

Lastly, the lack of standardized approaches to GNSS spoofing detection presents significant integration challenges across industries. Currently, there are no universally accepted metrics for spoofing detection performance or standard interfaces for reporting potential spoofing events. GPSwise by SkAI Data Services provides real-time GPS spoofing and jamming detection using ADS-B data from OpenSky Network, displaying global interference patterns on interactive maps for aviation safety (SkAI Data Services, n.d.). Stanford's Wide Area Augmentation System (WAAS) system offers GNSS interference detection through ADS-B data analysis, providing heatmaps of worldwide jamming events and monthly reports to enhance aviation navigation integrity (Stanford GPS Lab, n.d.).

Advanced Navigation (2024) reports that International Air Transport Association (IATA) has lately assembled global aviation leaders to confront the increasing spoofing threat, commencing a series of essential discussions that will influence geopolitical agendas moving forward, reflecting widespread anxiety across multiple industries. This fragmentation complicates both the development and deployment of consistent protection measures across critical infrastructure systems.

7.2 Computational Requirements and Deployment Feasibility

The practical deployment of GNSS timing spoofing detection systems is significantly influenced by computational requirements and resource constraints. This section examines the feasibility of implementing the proposed detection methods in real-world scenarios.

7.2.1 Computational Resource Analysis

The detection methods analysed in this thesis have varying computational demands that directly impact their deployment feasibility. Traditional detection methods based on signal characteristics and validity flags require minimal computational resources, making them suitable for integration into existing hardware with limited processing capabilities, while holdover oscillators serve as the primary mitigation tool by maintaining timekeeping when GNSS signals are compromised (Inside GNSS, 2020). These methods primarily rely on threshold-based detection of anomalies in readily available parameters such as C/N0 or pseudorange RMS values and position validity flags.

In contrast, the machine learning approach using the Isolation Forest algorithm presents more substantial computational requirements, particularly during the training phase. Real-time processing is essential for effective spoofing detection, with current advanced systems capable of processing data with latency of less than three seconds, crucial for critical infrastructure protection (GPSPATRON, 2023). Based on our implementation:

1. **Training Phase**: The Isolation Forest model training required moderate computational resources but was completed in acceptable timeframes on standard computing hardware. This process would typically be performed offline in a controlled environment rather than on the deployed receivers.

2. **Inference Phase**: The deployed model demonstrated efficient performance with minimal processing overhead. The analysis of approximately 4,700 instances was completed with negligible latency, suggesting feasibility for real-time applications.

3. **Memory Requirements**: The trained model's memory footprint was modest, requiring approximately less than 22 MB of storage space with additional weight files being less than a MB, making it suitable for deployment on most modern GNSS receivers or microcontrollers with sufficient processing capacity.

7.2.2 Scalability Considerations

Scaling detection systems across numerous receivers introduces additional computational challenges. For centralized architectures where data from multiple receivers is processed at a single point, bandwidth requirements and processing bottlenecks must be carefully considered, especially in large networks like those in telecommunications where 5G wireless infrastructure has more complex, higher-density synchronization needs (Inside GNSS, 2021). Based on our analysis of the Jammertest data, the receiver generated data of approximately 22 MB for a total of 1 hour, 27 minutes and 14 seconds. In interference scenarios this may even increase. Distributed detection architectures, where each receiver performs local analysis before reporting results, offer better scalability but require each node to have sufficient computational capability.

7.2.3 Power Consumption Implications

For battery-powered or energy-constrained applications, the power consumption of continuous spoofing detection presents a significant consideration. Since the GNSS signals received on the Earth are very weak (approximately around -130 dBm), equivalent to a 25W lightbulb seen from 16,000 km away, detection systems must be highly sensitive while maintaining efficient power consumption (Safran, n.d.). Since receiver specific validity flag methods is already built inside u-blox, it doesn't require any additional power beyond normal receiver operation, making them suitable for energy-constrained applications. Both the Isolation Forest algorithm and an algorithm that will check anomalies in pseudorange RMS error will require additional processing power during continuous operation compared to baseline receiver functionality, which may impact battery life in portable applications.

A tiered detection system can be built, where low-power traditional methods trigger more resource-intensive analysis only when suspicious conditions are detected. As demonstrated by validity flag, it starts to give false alarms before time jump which can be used as a switch to turn on the machine learning approach. It will offer a balanced

approach for energy-constrained scenarios, with something like a holdover oscillator maintaining timekeeping when GNSS is ignored (Inside GNSS, 2020).

7.2.4 Hardware Integration Feasibility

The feasibility of hardware integration varies significantly across different receiver types and applications. Modern receivers with programmable firmware can be supplied with additional suites of protection against GPS/GNSS cyber jamming and spoofing attacks, utilizing technology originally developed for military applications but now available for civilian critical infrastructure (AccuBeat, n.d.). Integration may require external processing modules that intercept and analyse GNSS data streams in case a receiver does not allow firmware updates. For this reason, we may have to include additional microcontroller such an Arduino or raspberry pi. This approach introduces additional hardware costs and potential points of failure but enables detection capabilities without replacing existing receivers.

7.2.5 Cost-benefit Analysis

The implementation costs must be weighed against the potential impact of timing spoofing attacks on critical infrastructure. The potential consequences of successful timing spoofing attacks could cause networks to "fall apart," with stock exchanges shutting down due to inability to reconcile trades, ATMs failing because banks can't verify funds, and eventually even affecting the electrical grid (Quartz, 2017). Based on our analysis:

1. **Implementation Costs**: The incremental cost of implementing traditional detection methods is minimal for modern receivers with programmable firmware. Machine learning approaches require more substantial initial investment in development and training but offer good detection performance as demonstrated by our results (99.96% specificity and 100% recall).

2. **Protection Value**: GPS World (2012) highlights that disruptions to GNSS services can have significant economic consequences, citing the 2003 Northeast blackout (though not spoofing-related) which resulted in roughly $6 billion in damages over just four days, underscoring the importance of safeguarding timing-dependent critical infrastructure. As mentioned in the chapter two, GNSS timing spoofing can also result in significant loss in power utilities, giving more reasons to implement a counter (Bin et al., 2020).

7.3 Potential Use cases and Industry Applications

The GNSS timing spoofing detection methods developed in this thesis have significant practical applications across various critical infrastructure sectors. This section explores the potential use cases and industry applications where these methodologies could provide substantial benefits.

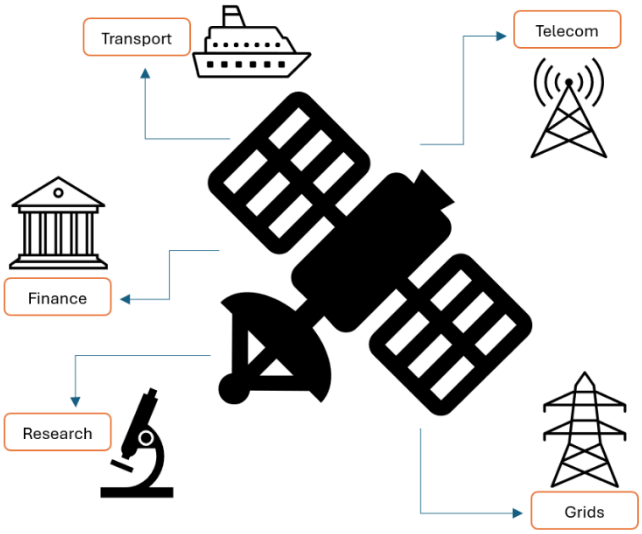

Figure 30: Illustration of potential industry applications

7.3.1 Telecommunications Networks

Telecommunications infrastructure represents one of the most timing-critical applications of GNSS technology. Modern cellular networks, including 5G implementations,

require precise timing from GNSS sources, with more complex, higher-density synchronization needs than previous-generation networks and high dependence on the integrity of live-sky GNSS timing signals (Inside GNSS, 2021). The detection methods developed in this research are particularly applicable to telecommunications infrastructure for several reasons:

1. **Base Station Synchronization**: The high accuracy and low false positive rate (0.04%) of the Isolation Forest model makes it suitable for protecting cellular base station timing systems where synchronization errors can cause handover failures and reduced network capacity.

2. **Network Resilience**: Critical time synchronization for utility, telecommunications, banking and computer industries has become increasingly dependent on GNSS signals, making robust detection systems essential for protecting these networks (Sathyamoorthy, 2013). The combination of traditional validity flag monitoring with machine learning detection creates a robust defence mechanism for protecting telecommunications backhaul networks.

3. **Cell Site Integration**: Security approaches that incorporate systems like BlueSky GNSS Firewall alongside synchronization supervision and administration platforms protect crucial timing infrastructure and 5G systems from GPS disruption and deception, while delivering integrated visibility through a single management interface (Inside GNSS, 2021). The moderate computational requirements of my proposed methods make them feasible for deployment directly at cell sites.

7.3.2 Power Grid Applications

As mentioned in the chapter 2, power grids require precise timing, making them the biggest client for our solution. The detection methodologies developed in this research offer several benefits for power grid applications:

1. **Substation Protection**: A large-scale electricity blackout can be caused by a targeted attack of a phasor measurement unit (PMU) which is used to control power grid, as these PMUs highly depend on precise timing provided by GPS systems (Syam, 2022) (Falletti et al., 2019). Our detection methods can be integrated into substation timing systems to prevent such attacks.

2. **Wide Area Monitoring**: Given the vital importance of power infrastructure and the expected reliance of smart grid technology on extremely accurate timing, ensuring GNSS signals can withstand interference, jamming, and spoofing is of great importance (GPSPATRON, 2019) (Zhang et al, 2020). The high specificity (99.96%) of our machine learning approach enables reliable detection across geographically distributed power infrastructure.

3. **Smart Grid Security**: According to EUSPA (2020), GNSS receivers provide relatively inexpensive, dependable, and highly precise timing capabilities that can be deployed extensively throughout smart grids for instantaneous automated control, with contemporary systems incorporating multiple defensive measures against spoofing. As power grids evolve toward more distributed architectures, the timing security provided by these detection methods becomes increasingly valuable.

7.3.3 Financial Services Infrastructure

Financial transactions and trading systems rely on precise timing for transaction sequencing, audit trails, and compliance with regulatory requirements. The lack of accurate GNSS timing will render everyday financial and other activities impossible such as making phone calls, stock purchases, ATM withdrawals, or electricity consumption, with financial firms possibly needing to suspend operations if they are unable to confirm trades (Quartz, 2017). The detection methods developed in this research can be applied to financial infrastructure in several ways:

1. **Trading Platform Security**: Platforms for high-frequency trading, where traders have invested millions to enhance algorithms and communication networks for executing transactions microseconds ahead of rivals, rely on exact consensus regarding the precise timing of each trade (Quartz, 2017). My detection methods provide robust protection against timing manipulation.

2. **Transaction Validation**: Fake time introduced by GNSS spoofing can disrupt financial transactions and compromise financial data, causing severe economic impacts and affecting reliability of banking systems and stock exchanges (GPSPATRON, 2023). The detection methods can help ensure the integrity of timestamp-based transaction validation systems.

3. **Distributed Ledger Applications**: Blockchain and other distributed ledger technologies rely on accurate timing for consensus mechanisms, making them potential beneficiaries of enhanced timing security through our combined traditional and AI-based detection approaches.

7.3.4 Transportation and Navigation Systems

Beyond stationary infrastructure, the detection methods also have applications in mobile scenarios. From September 2023 onward, electronic combat in conflict regions has intensified spoofing events, with IATA revealing that Airbus logged close to 50,000 interference occurrences in 2022, which marks a four-time increase over the previous year's figures (Advanced Navigation, 2024). Even though timing doesn’t directly affect them, they can still benefit from having resilience to timing spoofing. So based on that, this research applies to:

1. **Maritime Navigation**: Shipping vessels are increasingly subjected to spoofing attempts, especially in Baltic Sea areas, where they are being diverted from their intended paths into territorial waters, underscoring the importance of

implementing interference countermeasures that utilize contemporary security approaches (GIM International, 2021) (Kauranen, 2024).

2. **Aviation Systems**: Aircraft navigation systems and ground-based aviation infrastructure require high-integrity timing and positioning information, particularly for precision approach and landing procedures, where our detection methods could provide crucial protection.

3. **Autonomous Vehicles**: Tesla's Model S and Model 3 were demonstrated to be susceptible to navigation spoofing in 2019, when researchers showed how attackers could manipulate the autopilot system to guide vehicles off intended roadways, accentuating the critical need for robust detection technologies in autonomous driving platforms (CybersecAsia, 2022).

7.3.5 Scientific and Research Applications

The methodologies developed also have applications in scientific and research contexts:

1. **Timing-Based Research**: Scientific experiments requiring precise timing synchronization, such as distributed seismic arrays or radio astronomy, can benefit from enhanced protection against timing manipulation through our combined traditional and AI-based approach.

2. **Environmental Monitoring**: Distributed sensor networks for environmental monitoring rely on accurate and secure time, with modern solutions using high-quality timing GNSS receivers with interference mitigation technology to withstand malicious or accidental RF jamming and spoofing (Septentrio, n.d.).

The successful implementation of the detection methodologies across these sectors would significantly enhance the resilience of critical infrastructure against GNSS timing spoofing attacks. According to Meng et al. (2022), the research focus has transitioned

from enhancing positioning precision to broadening system applications and optimizing performance, with satellite navigation systems' resilience against spoofing emerging as a prominent research area within the navigation domain. The combination of traditional signal-based detection with machine learning approaches offers a balanced solution that addresses the diverse requirements of these applications while maintaining practical feasibility for real-world deployment.

8 Conclusions and Future Work

This thesis presented an analysis of GNSS timing spoofing detection methodologies using data from Jammertest 2024, giving the literature review on exploring the fundamentals of GNSS timing synchronization, existing spoofing techniques, and the evolution of detection methodologies from traditional signal-based approaches to advanced AI implementations. The research examined how various GNSS parameters, including raw measurements (pseudoranges, carrier phase observations etc) and calculated metrics (position, power of spoofer, HDOP values), behaved during timing spoofing attacks. Both conventional detection methods and a machine learning algorithm were implemented and benchmarked to identify effective spoofing detection strategies. Analysis of the raw GNSS observations revealed that while carrier-to-noise ratio (C/N0) measurements did not provide reliable indications of spoofing, other traditional methods including pseudorange RMS error monitoring and validity flags successfully detected timing anomalies. These methods demonstrated effective detection capabilities but were prone to false positives in the data analysed.

The machine learning approach using the Isolation Forest algorithm showed good performance with 100% recall and 99.96% specificity, correctly identifying all spoofing instances while generating only two false positives out of 4,695 normal samples. This high accuracy demonstrates the potential of unsupervised anomaly detection for identifying timing spoofing attacks without requiring extensive labelled training data. The combined approach of using validity flags as an initial trigger mechanism followed by machine learning verification is foreseen to be a robust strategy for balancing detection sensitivity with computational efficiency.

Several avenues for future research have been identified. First and foremost, the limited spoofing data available (only 1.88 minutes of actual spoofed reception) necessitates expanded data collection to further validate and refine the detection methodologies. A more diverse dataset including different spoofing scenarios, receiver types, and environmental conditions would improve model robustness.

Exploration of alternative machine learning approaches, particularly reinforcement learning techniques which are currently underrepresented in GNSS spoofing detection literature, could yield additional performance improvements. Such techniques could adapt to evolving spoofing strategies and optimize the trade-off between detection accuracy and false alarm rates.

For practical industry applications, model optimization using TinyML techniques could significantly reduce computational requirements, enabling implementation on resource-constrained embedded systems commonly used in critical infrastructure. This would make the solution accessible to a wider range of applications without requiring hardware upgrades.

Finally, integrating the detection methodologies with automated mitigation strategies, such as holdover oscillator switching or alternative timing source selection, would create a complete protective system for critical infrastructure dependent on accurate timing.

References

AccuBeat. (n.d.). GNSS Cyber Security & Anti-Spoofing Products. *AccuBeat*. Retrieved May 7, 2025, from https://www.accubeat.com/cybersecurity-antispoofing-products

Advanced Navigation. (2024, February 14). Navigating the Rising Threat of GNSS Spoofing in Critical Industries. *Advanced Navigation*. Retrieved May 7, 2025, from https://www.advancednavigation.com/tech-articles/navigating-the-rising-threat-of-gnss-spoofing-in-critical-industries/

Agafonkin, V. (2016). Fast geodesic approximations with Cheap Ruler. *Mapbox*. Retrieved April 12, 2025, from https://blog.mapbox.com/fast-geodesic-approximations-with-cheap-ruler-106f229ad016

Aggarwal, V., Gupta, V., Singh, P., Sharma, K., & Sharma, N. (2019). Detection of spatial outlier by using improved Z-score test. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 788-790). IEEE. https://doi.org/10.1109/ICOEI.2019.8862582

Angrisano, A., Gaglione, S., Gioia, C., Borio, D., & Fortuny-Guasch, J. (2013). Testing the test satellites: The Galileo IOV measurement accuracy. In *2013 International Conference on Localization and GNSS (ICL-GNSS)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICL-GNSS.2013.6577253

ArduSimple. (2024). Product summary: SimpleRTK2B V3. *ArduSimple*. Retrieved April 11, 2025, from https://www.ardusimple.com/wp-admin/admin-post.php?action=generate_product_pdf&product_id=35134

Azdy, R. A., & Darnis, F. (2020). Use of Haversine formula in finding distance between temporary shelter and waste end processing sites. *Journal of Physics: Conference Series*, 1500, 012121. https://doi.org/10.1088/1742-6596/1500/1/012104

Behrendt, K., & Fodero, K. (2006, May). The perfect time: An examination of time-synchronization techniques. Paper presented at the 60th Annual Georgia Tech Protective Relaying Conference, Atlanta, GA, USA, 3–5 May 2006. Retrieved May 5, 2025, from https://selinc.com/api/download/3686

Bin, Q., Ziwen, C., Yong, X., Liang, H., & Sheng, S. (2020). GPS spoofing-based time synchronisation attack in advanced metering infrastructure and its protection. *The Journal of Engineering*, 2020(9), 809-815. https://doi.org/10.1049/joe.2020.0022

Cabinet Office. (2025). Overview of the Quasi-Zenith Satellite System (QZSS). *QZSS.go.jp*. Retrieved May 22, 2025, from https://qzss.go.jp/en/overview/services/sv01_what.html

Cao, H., Shen, J., Yan, R., Zhao, Y., Xu, W., & Geng, L. (2024). More robust high precision time synchronization system. *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication*, 1-6. https://doi.org/10.1109/ISPCS63021.2024.10747722

Chandler, D. (2022, December 5). The role of atomic clocks in data centers. *GPS World*. https://www.gpsworld.com/the-role-of-atomic-clocks-in-data-centers/

Chang, H., Pang, C., Zhang, L., & Guo, Z. (2022). Rotating single-antenna spoofing signal detection method based on IPNN. *Sensors*, 22(19), 7141. https://doi.org/10.3390/s22197141

Chen, J., Wang, X., Fang, Z., Jiang, C., Gao, M., & Xu, Y. (2024). A real-time spoofing detection method using three low-cost antennas in satellite navigation. *Electronics*, 13(6), 1134. https://doi.org/10.3390/electronics13061134

Chen, S., Ni, S., Lei, T., Cheng, L., & Song, X. (2024). GNSS spoofing detection via the intersection angle between two directions of arrival in a single rotating antenna. *Sensors*, 24(4), 1116. https://doi.org/10.3390/s24041116

CybersecAsia. (2022, January 20). Cyberattacks on satellite signals a growing GNSS threat. *CybersecAsia*. Retrieved May 7, 2025, from https://cybersecasia.net/tips/cyberattacks-on-satellite-signals-a-growing-gnss-threat/

Dasgupta, S., Ghosh, T., & Rahman, M. (2022). A reinforcement learning approach for global navigation satellite system spoofing attack detection in autonomous vehicles. *Transportation Research Record*, 2676(12), 318-330. https://doi.org/10.1177/03611981221095509

European GNSS Service Centre. (2025). Galileo Open Service Navigation Message Authentication (OSNMA). *European GNSS Service Centre*. Retrieved April 10, 2025,

from https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS-NMA_Info_Note.pdf

European Space Agency. (2011). GNSS all signals [Image]. *Navipedia*. Retrieved May 22, 2025, from https://gssc.esa.int/navipedia/index.php?title=File:GNSS_All_Signals.png

European Space Agency. (n.d.). How the Galileo atomic clocks work. *European Space Agency*. Retrieved May 5, 2025, from https://www.esa.int/Applications/Navigation/How_the_Galileo_atomic_clocks_work

EUSPA. (2020, December 15). Timing is everything – GNSS and the energy grids of the future. *European GNSS Service Centre*. Retrieved May 7, 2025, from https://www.euspa.europa.eu/newsroom-events/news-archive/timing-everything-gnss-and-energy-grids-future

EUSPA. (n.d.). What is Galileo? *GSC-Europa.eu*. Retrieved May 20, 2025, from https://www.gsc-europa.eu/galileo/what-is-galileo

Falletti, E., Margaria, D., Marucco, G., Motella, B., Nicola, M., & Pini, M. (2019). Synchronization of critical infrastructures dependent upon GNSS: Current vulnerabilities and protection provided by new signals. *IEEE Systems Journal*, 13(3), 2118-2129. https://doi.org/10.1109/JSYST.2018.2883752

Feng, S., Ochieng, W., Samson, J., et al. (2012). Integrity monitoring for carrier phase ambiguities. *Journal of Navigation*, 65(1), 41–58. https://doi.org/10.1017/S037346331100052X

Finance Derivative. (2024). GNSS vulnerabilities: Securing the future of finance with new PNT solutions. *Finance Derivative*. Retrieved May 2, 2025, from https://www.financederivative.com/gnss-vulnerabilities-securing-the-future-of-finance-with-new-pnt-solutions/

Fingrid. (2024). Prospects for future electricity production and consumption: Fingrid's forecast Q3/2024 [PDF]. *Fingrid Oyj*. Retrieved May 5, 2025, from https://www.fingrid.fi/globalassets/dokumentit/en/news/prospects-for-future-electricity-prodution-and-consumption.-fingrids-forecast-q3-2024.pdf

Fourati, F., & Alouini, M.-S. (2021). Artificial intelligence for satellite communication: A review. *Intelligent and Converged Networks*, 2(3), 213-244. IEEE. http://doi.org/10.23919/ICN.2021.0015

Gao, W., Li, H., Zhong, M., & Lu, M. (2023). The separate clock drift matched filter to detect time synchronization attacks toward global navigation satellite systems. *IEEE Transactions on Industrial Electronics*, 70(6), 6305-6315. https://doi.org/10.1109/TIE.2022.3194578

Gao, Y., & Li, G. (2022). Three time spoofing algorithms for GNSS timing receivers and performance evaluation. *GPS Solutions*, 26(87). https://doi.org/10.1007/s10291-022-01275-7

GIM International. (2021, March 8). What is GNSS Spoofing?. *GIM International*. Retrieved May 7, 2025, from https://www.gim-international.com/content/article/what-is-gnss-spoofing

Google Developer. (2025, March 3). Classification: Accuracy, recall, precision, and related metrics. *Google for Developers*. Retrieved April 28, 2025, from https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

GPS World. (2012, August 1). Going Up Against Time: The Power Grid's Vulnerability to GPS Spoofing Attacks. *GPS World*. Retrieved May 7, 2025, from https://www.gpsworld.com/wirelessinfrastructuregoing-against-time-13278/

GPSPATRON. (2019, April 23). The Power Grid's Vulnerability to GPS Spoofing Attacks. *GPSPATRON*. Retrieved May 7, 2025, from https://gpspatron.com/power-grid-spoofing/

GPSPATRON. (2023, January 31). GNSS Interference Monitoring and Classification for Critical Infrastructure Safety. *GPSPATRON*. Retrieved May 7, 2025, from https://gpspatron.com/gnss-interference-monitoring-and-classification-for-critical-infrastructure-safety/

Hasan, K. F., Feng, Y., & Tian, Y. C. (2023). Precise GNSS Time Synchronization With Experimental Validation in Vehicular Networks. *IEEE Transactions on Network and*

Service Management, 20(3), 3289-3301. https://doi.org/10.1109/TNSM.2022.3228078

He, C., Lu, X., Guo, J., Su, C., Wang, W., & Wang, M. (2020). Initial analysis for characterizing and mitigating the pseudorange biases of BeiDou navigation satellite system. *Satellite Navigation*, 1(3). https://doi.org/10.1186/s43020-019-0003-3

Hollberg, L. (2021). Atomic clocks for GNSS. In *Position, navigation, and timing technologies in the 21st century: Integrated satellite navigation, sensor systems, and civil applications* (pp. 1497–1519). IEEE. https://doi.org/10.1002/9781119458555.ch47

Honkala, S., Thombre, S., Kirkko-Jaakkola, M., Zelle, H., Veerman, H., Wallin, A. E., Dierikx, E. F., Kaasalainen, S., Söderholm, S., & Kuusniemi, H. (2020). Performance of EGNSS-based timing in various threat conditions. *IEEE Transactions on Instrumentation and Measurement*, 69(5), 2287–2299. https://doi.org/10.1109/TIM.2019.2923485

Hu, H., & Fang, L. (2009). GPS cycle slip detection and correction based on high order difference and Lagrange interpolation. *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, 384-387. https://doi.org/10.1109/PEITS.2009.5406991

IAC. (n.d.). About GLONASS. *GLONASS-IAC.ru*. Retrieved May 22, 2025, from https://glonass-iac.ru/en/about_glonass/

Inside GNSS. (2014). Financial networks shifting to GPS-stamped precise time. *Inside GNSS*. Retrieved May 2, 2025, from https://insidegnss.com/financial-networks-shifting-to-gps-stamped-precise-time/

Inside GNSS. (2019). GPS combined with Galileo to provide robust timing for the financial sector. *Inside GNSS*. Retrieved May 2, 2025, from https://insidegnss.com/gps-combined-with-galileo-to-provide-robust-timing-for-the-financial-sector/

Inside GNSS. (2020, April 8). State of Play: Resilient Timekeeping for Critical Infrastructure. *Inside GNSS*. Retrieved May 7, 2025, from https://insidegnss.com/state-of-play-resilient-timekeeping-for-critical-infrastructure/

Inside GNSS. (2021, June 24). Timing Sources for Wireless Carriers, Critical Infrastructure Bolstered Against GNSS Jamming and Spoofing. *Inside GNSS*. Retrieved May 7, 2025, from https://insidegnss.com/timing-sources-for-wireless-carriers-critical-infrastructure-bolstered-against-gnss-jamming-and-spoofing/

Iqbal, A., Aman, M. N., & Sikdar, B. (2023-a). Machine learning based time synchronization attack detection for synchrophasors. *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2251-2256. https://doi.org/10.1109/GLOBECOM54140.2023.10437566

Iqbal, A., Aman, M. N., & Sikdar, B. (2023-b). Representation learning based time synchronization attack detection for synchrophasors. *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 1-6. https://doi.org/10.1109/SmartGridComm57358.2023.10333911

ISRO. (2023). Satellite navigation services. *ISRO.gov.in*. Retrieved May 22, 2025, from https://www.isro.gov.in/SatelliteNavigationServices.html

Jammertest. (2024-a). Test logs from Jammertest 2024. *Jammertest*. Retrieved April 10, 2025, from https://jammertest.no/content/files/2025/02/Logg_Jammertest_2024_v1.xlsx

Jammertest. (2024-b). Jammertest 2024 Test Catalog. *Jammertest*. Retrieved April 10, 2025, from https://jammertest.no/content/files/2025/02/Testcatalog.pdf

Jammertest. (2025). About Jammertest. *Jammertest*. Retrieved April 9, 2025, from https://jammertest.no/about/

Jia, P., & Liao, Q. (2025). GNSS Time Synchronization Attack detection algorithm based on clock bias change covariance. *2025 IEEE 5th International Conference on Power, Electronics and Computer Applications (ICPECA)*, 853-857. https://doi.org/10.1109/ICPECA63937.2025.10928718

Jianfeng, W., Qishan, Z., Jie, S., Qin, W., & Xiangwei, Z. (2016). Design and implementation of GNSS time service system. *IEEE*. https://doi.org/10.1109/ICCS.2016.7889263

Jin, S. (2012). *Global Navigation Satellite Systems*. InTech.

Kaplan, E. D., & Hegarty, C. J. (2006). *Understanding GPS: Principles and applications* (2nd ed.). Artech House.

Kauranen, A. (2024, October 31). Finland detects satellite navigation jamming and spoofing in Baltic Sea. *Reuters*. Retrieved May 7, 2025, from https://www.reuters.com/world/europe/finland-detects-satellite-navigation-jamming-spoofing-baltic-sea-2024-10-31/

Khoei, T. T., Gasimova, A., Ahajjam, M. A., Shamaileh, K. A., Devabhaktuni, V., & Kaabouch, N. (2022). A comparative analysis of supervised and unsupervised models for detecting GPS spoofing attack on UAVs. *2022 IEEE International Conference on Electro Information Technology (eIT)*, 279-284. https://doi.org/10.1109/eIT53891.2022.9813826

Lee, D., Miralles, D., Akos, D., Konovaltsev, A., Kurz, L., Lo, S., & Nedelkov, F. (2020). Detection of GNSS spoofing using NMEA messages. *2020 European Navigation Conference (ENC)*, 1-10. https://doi.org/10.23919/ENC48637.2020.9317470

Lee, J., Schmidt, E., Gatsis, N., & Akopian, D. (2023). Detection and mitigation of spoofing attacks against time synchronization and positioning. *IEEE Access*, 11, 138986-139003. https://doi.org/10.1109/ACCESS.2023.3341028

Li, J., Chen, Z., Yuan, X., Xie, T., Xu, Y., Zheng, Z., & Zhu, X. (2025). A real-time GNSS time spoofing detection framework based on feature processing. *GPS Solutions*, 29(45). https://doi.org/10.1007/s10291-024-01802-8

Li, J., Li, W., He, S., Dai, Z., & Fu, Q. (2020). Research on detection of spoofing signal with small delay based on KNN. In *2020 IEEE 3rd International Conference on Electronics Technology (ICET)* (pp. 625-629). IEEE. https://doi.org/10.1109/ICET49382.2020.9119515

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Liu, R., Yang, Z., Chen, Q., Liao, G., & Zhu, Q. (2023). Localization of GNSS spoofing interference source based on a moving array antenna. *Remote Sensing*, 15(23), 5497. https://doi.org/10.3390/rs15235497

Ma, X., Gao, M., Zhao, Y., & Yu, M. (2024). A novel navigation spoofing algorithm for UAV based on GPS/INS-integrated navigation. *IEEE Transactions on Vehicular Technology*, 73(10), 15424-15439. https://doi.org/10.1109/TVT.2024.3401856

Ma, Y., Li, H., Zhou, Z., et al. (2024). C/N0 estimation based on acquisition correlation ratio for short GNSS data. *GPS Solutions*, 28, 143. https://doi.org/10.1007/s10291-024-01666-y

Mao, P., Yuan, H., Chen, X., Gong, Y., Li, S., Li, R., Luo, R., Zhao, G., Fu, C., & Xu, J. (2023). A GNSS spoofing detection and direction-finding method based on low-cost commercial board components. *Remote Sensing*, 15(11), 2781. https://doi.org/10.3390/rs15112781

Meng, L., Yang, L., Yang, W., & Zhang, L. (2022). A Survey of GNSS Spoofing and Anti-Spoofing Technology. *Remote Sensing*, 14(19), 4826. https://doi.org/10.3390/rs14194826

Mohammed, A. B., Fourati, L. C., & Fakhrudeen, A. M. (2024). Isolation Forest algorithm against UAV's GPS spoofing attack. In *2024 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics* (pp. 459-463). IEEE. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics62450.2024.00090

Moritz, H. (2000). Geodetic Reference System 1980. *Journal of Geodesy*, 74(1), 128-133. Retrieved from https://link.springer.com/article/10.1007/s001900050278

NASA. (2019). What is an atomic clock? *NASA*. Retrieved May 5, 2025, from https://www.nasa.gov/missions/tech-demonstration/deep-space-atomic-clock/what-is-an-atomic-clock/

NCO. (2021, February 22). GPS overview. *GPS.gov*. Retrieved May 22, 2025, from https://www.gps.gov/systems/gps/

NCO. (2022). Timing applications. *GPS.gov*. Retrieved May 5, 2025, from https://www.gps.gov/applications/timing/

NIST. (n.d.). A brief history of atomic time. *National Institute of Standards and Technology*. Retrieved May 5, 2025, from https://www.nist.gov/atomic-clocks/brief-history-atomic-time

NovAtel. (n.d.-a). GPGGA – GPS fix data and undulation. *NovAtel*. Retrieved April 20, 2025, from https://docs.novatel.com/OEM7/Content/Logs/GPGGA.htm

NovAtel. (n.d.-b). GPRMC – GPS specific information. *NovAtel*. Retrieved April 20, 2025, from https://docs.novatel.com/OEM7/Content/Logs/GPRMC.htm

NovAtel. (n.d.-c). GLONASS (Global Navigation Satellite System), Russia. *NovAtel*. Retrieved May 5, 2025, from https://novatel.com/an-introduction-to-gnss/gnss-constellations/glonass

NovAtel. (n.d.-d). Galileo (European Union). *NovAtel*. Retrieved May 5, 2025, from https://novatel.com/an-introduction-to-gnss/gnss-constellations/galileo

NovAtel. (n.d.-e). BeiDou (China). *NovAtel*. Retrieved May 5, 2025, from https://novatel.com/an-introduction-to-gnss/gnss-constellations/beidou

NovAtel. (n.d.-f). QZSS (Quasi-Zenith Satellite System), Japan. *NovAtel*. Retrieved May 5, 2025, from https://novatel.com/an-introduction-to-gnss/gnss-constellations/qzss

NovAtel. (n.d.-g). NavIC (Navigation with Indian Constellation), India. *NovAtel*. Retrieved May 5, 2025, from https://novatel.com/an-introduction-to-gnss/gnss-constellations/navic

Pearson, J. (2024, August 11). GPS spoofers 'hack time' on commercial airlines, researchers say. *Reuters*. Retrieved April 24, 2025, from https://www.reuters.com/technology/cybersecurity/gps-spoofers-hack-time-commercial-airlines-researchers-say-2024-08-10/

Perrig, A., Canetti, R., Tygar, J. D., & Song, D. (2002). The TESLA Broadcast Authentication Protocol. *CryptoBytes*, 5(2), 2-13. Retrieved from https://people.eecs.berkeley.edu/~tygar/papers/TESLA_broadcast_authentication_protocol.pdf

Psiaki, M. L., & Humphreys, T. E. (2016). GNSS spoofing and detection. *Proceedings of the IEEE*, 104(6), 1258-1270. https://doi.org/10.1109/JPROC.2016.2526658

Psiaki, M., & Humphreys, T. (2021). Civilian GNSS spoofing, detection, and recovery. In *Position, navigation, and timing technologies in the 21st century: Integrated satellite navigation, sensor systems, and civil applications* (pp. 655-680). IEEE. https://doi.org/10.1002/9781119458449.ch25

Qian, B., Cai, Z., Xiao, Y., Hong, L., & Su, S. (2020). GPS spoofing-based time synchronisation attack in advanced metering infrastructure and its protection. *The Journal of Engineering*, 2020(9), 809-815. https://doi.org/10.1049/joe.2020.0022

Quartz. (2017). The GPS/GNSS system behind finance, telecommunications and transportation networks is vulnerable to terrorist jamming and criminal spoofing. *Quartz*. Retrieved May 5, 2025, from https://qz.com/1106064/the-entire-global-financial-system-depends-on-gps-and-its-shockingly-vulnerable-to-attack

Radoš, K., Brkić, M., & Begušić, D. (2024). Recent advances on jamming and spoofing detection in GNSS. *Sensors*, 24(13), 4210. https://doi.org/10.3390/s24134210

Ranganathan, A., Belfki, A., & Closas, P. (2023). Analyzing the impact of GNSS spoofing on the formation of unmanned vehicles swarms. In *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)* (pp. 3138–3147). The Institute of Navigation. https://doi.org/10.33012/2023.19438

Romaniuc, A.-G., Vasile, V.-C., Borda, M.-E., & Alexandru, B. (2024). NTP spoofing attack detection on time servers with GNSS sensors based on Long Short-Term Memory algorithm. *2024 15th International Conference on Communications (COMM)*, 1-6. https://doi.org/10.1109/COMM62355.2024.10741488

Rouan, D. (2023). Doppler shift. In Gargaud, M., et al. (Eds.), *Encyclopedia of Astrobiology*. Springer. https://doi.org/10.1007/978-3-662-65093-6_455

Ruiqiong, C., Ya, L., Xiaohui, L., Duosheng, F., & Ying, Y. (2019). High-precision time synchronization based on common performance clock source. *IEEE International Conference on Electronic Measurement & Instruments*, 1363-1368. https://doi.org/10.1109/ICEMI46757.2019.9101703

Rustamov, A., Gogoi, N., Minetto, A., & Dovis, F. (2020). Assessment of the vulnerability to spoofing attacks of GNSS receivers integrated in consumer devices. In *2020*

International Conference on Localization and GNSS (ICL-GNSS) (pp. 1-6). IEEE. https://doi.org/10.1109/ICL-GNSS49876.2020.9115489

Safran. (2024, August 6). Measuring a GNSS Signal and Gaussian Noise Power. *Safran*. Retrieved May 7, 2025, from https://safran-navigation-timing.com/document/measuring-a-gnss-signal-and-gaussian-noise-power/

Sathyamoorthy, D. (2013). Global navigation satellite system (GNSS) spoofing: A review of growing risks and mitigation steps. *Defence S&T Technical Bulletin*, 6(1), 42-61. Science & Technology Research Institute for Defence (STRIDE). Retrieved May 7, 2025, from https://www.researchgate.net/profile/Dinesh-Sathyamoorthy/publication/259465910_Global_navigation_satellite_system_GNSS_spoofing_A_review_of_growing_risks_and_mitigation_steps/links/00b4952bdc288d6301000000/Global-navigation-satellite-system-GNSS-spoofing-A-review-of-growing-risks-and-mitigation-steps.pdf

semuadmin. (2025). pyubx2 - PyPI. *PyPI*. Retrieved April 13, 2025, from https://pypi.org/project/pyubx2/

Septentrio. (n.d.). Insight: GPS delivers secure time for critical infrastructure. *Septentrio*. Retrieved May 7, 2025, from https://www.septentrio.com/en/learn-more/insights/how-gps-helps-keeping-time-heartbeat-connected-society

Shafique, A., Mehmood, A., & Elhadef, M. (2021). Detecting signal spoofing attack in UAVs using machine learning models. *IEEE Access*, 9, 93803-93815. https://doi.org/10.1109/ACCESS.2021.3089847

Shang, X., Sun, F., Zhang, L., Zhang, Q., Tang, X., & Gao, C. (2022). Detection and mitigation of GNSS spoofing via the pseudorange difference between epochs in a multi-correlator receiver. *GPS Solutions*, 26(2), 37. https://doi.org/10.1007/s10291-022-01224-4

Shereen, E., Ramakrishna, R., & Dán, G. (2022). Detection and localization of PMU time synchronization attacks via graph signal processing. *IEEE Transactions on Smart Grid*, 13(4), 3241-3254. https://doi.org/10.1109/TSG.2022.3150954

SkAI Data Services. (n.d.). GPSwise: Live GPS spoofing & jamming tracker map. *Spoofing.skai-data-services.com*. Retrieved May 22, 2025, from https://spoofing.skai-data-services.com/

SKYbrary Aviation Safety. (2021–2025). GNSS jamming and spoofing. *SKYbrary*. Retrieved April 24, 2025, from https://skybrary.aero/articles/gnss-jamming-and-spoofing

Spravil, J., Hemminghaus, C., von Rechenberg, M., Padilla, E., & Bauer, J. (2023). Detecting maritime GPS spoofing attacks based on NMEA sentence integrity monitoring. *Journal of Marine Science and Engineering*, 11(5), 928. https://doi.org/10.3390/jmse11050928

Stanford GPS Lab. (n.d.). GNSS interference detection using ADS-B. *WAAS-NAS.stanford.edu*. Retrieved May 22, 2025, from https://waas-nas.stanford.edu/

State Council Information Office of the People's Republic of China. (2016). *China's BeiDou navigation satellite system*. Foreign Languages Press. Retrieved May 22, 2025, from http://en.beidou.gov.cn/SYSTEMS/WhitePaper/201806/P020180608507822432019.pdf

Syam, W. (2022, July 18). GNSS spoofing: a fatal attack on GNSS system that is difficult to detect. *WASY Research*. Retrieved May 7, 2025, from https://www.wasyresearch.com/gnss-spoofing-a-fatal-attack-on-gnss-system-that-is-difficult-to-detect/

Tavotech. (n.d.). GPS NMEA sentence structure. *Tavotech*. Retrieved April 13, 2025, from https://tavotech.com/gps-nmea-sentence-structure/

Tondaś, D., Ilieva, M., van Leijen, F., & van der Marel, H. (2023). Kalman filter-based integration of GNSS and InSAR observations for local nonlinear strong deformations. *Journal of Geodesy*, 97, Article 109. https://doi.org/10.1007/s00190-023-01789-z

u-blox. (2024). ZED-F9P-02B data sheet. *u-blox*. Retrieved April 12, 2025, from https://www.mouser.fi/datasheet/2/1025/ZED_F9P_02B_DataSheet_UBX_21023276-3180703.pdf

Upadhyay, A. (2019). Haversine formula – Calculate geographic distance on earth. *IGISMAP*. Retrieved April 11, 2025, from https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/

Wang, J., Tang, X., Ma, P., Wu, J., Ma, C., & Sun, G. (2023). GNSS spoofing detection using Q channel energy. *Remote Sensing*, 15(22), 5337. https://doi.org/10.3390/rs15225337

Wei, X., & Sikdar, B. (2019). Impact of GPS time spoofing attacks on cyber physical systems. *2019 IEEE International Conference on Industrial Technology (ICIT)*, 1155-1160. https://doi.org/10.1109/ICIT.2019.8755016

Wei, X., Aman, M. N., & Sikdar, B. (2022). Exploiting correlation among GPS signals to detect GPS spoofing in power grids. *IEEE Transactions on Industry Applications*, 58(1), 697-708. https://doi.org/10.1109/TIA.2021.3131970

Xiao, L., Li, X., & Wang, G. (2019). GNSS spoofing detection using pseudo-range double differences between two receivers. In *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)* (pp. 498-502). IEEE. https://doi.org/10.1109/ICCSNT47585.2019.8962453

Xie, J., Liu, Q., Wang, L., Gong, Y., & Zhang, Z. (2022). Localizing GNSS spoofing attacks using direct position determination. *IEEE Sensors Journal*, 22(15), 15323-15333. https://doi.org/10.1109/JSEN.2022.3179557

Yang, B., Tian, M., Ji, Y., Cheng, J., Xie, Z., & Shao, S. (2022). Research on GNSS spoofing mitigation technology based on spoofing correlation peak cancellation. *IEEE Communications Letters*, 26(12), 3024-3028. https://doi.org/10.1109/LCOMM.2022.3204944

Yang, H., Jin, R., Xu, W., Che, L., & Zhen, W. (2023). Satellite navigation spoofing interference detection and direction finding based on array antenna. *Sensors*, 23(3), 1604. https://doi.org/10.3390/s23031604

Yuanfa, J., Xigang, S., & Huli, S. (2009). The pseudorange measurement equations and their solution for satellite navigation. In *2009 IEEE International Conference on Automation and Logistics* (pp. 1710–1715). https://doi.org/10.1109/ICAL.2009.5262698

Zhang, H., Peng, S., Liu, L., Su, S., & Cao, Y. (2020). Review on GPS spoofing-based time synchronisation attack on power system. *IET Generation, Transmission & Distribution*, 14(20), 4301-4309. https://doi.org/10.1049/iet-gtd.2020.0253

Zhang, Z., & Zhan, X. (2018). Statistical analysis of spoofing detection based on TDOA. *IEEJ Transactions on Electrical and Electronic Engineering*, 13(6), 840-850. https://doi.org/10.1002/tee.22637